

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Goran Popović

Autonomous agent localization in dynamic scenarios based on visual sensor data fusion

DOCTORAL THESIS

Zagreb, 2022



University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Goran Popović

Autonomous agent localization in dynamic scenarios based on visual sensor data fusion

DOCTORAL THESIS

Supervisor: Academic Professor Ivan Petrović, PhD

Zagreb, 2022



Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Goran Popović

Lokalizacija autonomnoga agenta u dinamičnim scenarijima zasnovana na fuziji podataka vizualnih senzora

DOKTORSKI RAD

Mentor: akademik prof. dr. sc. Ivan Petrović

Zagreb, 2022.

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

Supervisor: Academic Professor Ivan Petrović, PhD

Thesis contains 113 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 more than 40 national and 25 international scientific projects, where from them ten 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 about 80 papers in scientific journals and 220 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), award of the Croatian Academy of Sciences and Arts (2017), award "Nikola Tesla" from IEEE Croatia Section (2019) and award of the City of Zagreb (2022).

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 više od 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 oko 80 znanstvenih radova u časopisima i 220 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, 2017. godine nagradu Hrvatske akademije znanosti i umjetnosti, 2019. godine nagradu „Nikola Tesla" Hrvatske sekcije IEEE te 2022. godine nagradu Grada Zagreba.

ZAHVALA

Na kraju zadnjeg segmenta puta koji predstavlja moje akademsko obrazovanje želio bih se zahvaliti svima koji su direktno i indirektno na njega utjecali.

Najprije bih se želio zahvaliti mentoru akademiku prof. dr. sc. Ivanu Petroviću što mi je omogućio da se tijekom doktorata mogu neometano posvetiti istraživačkom radu. Također, zahvalio bih se mentoru i izv. prof. Ivanu Markoviću što su svojim konstruktivnim komentarima pomogli usmjeriti moje istraživanje u područje koje je rezultiralo konačnim znanstvenim doprinosima.

Zahvalio bih se prijateljima i kolegama iz LAMORA koji su pauze između rada na doktoratu učinili zabavnim, cimerima Josipu, Petkiju, Bokiju, Anđeli, Demijanu, Igoru, Ivanu i Vlahi, te Mariji, Antei, Jeleni, Ani, Marti, Juri, Luki, Filipu i Karlu s kojima nisam imao priliku biti cimer. Također, zahvalio bih se i prijateljima Nicole, Uni, Hrabaru, Lokotaru, Hrvoju, Danku i Žoržu s kojima sam često ručao i družio se u slobodno vrijeme.

Posebno se moram zahvaliti Damjanu, Goranu i Predragu koji su mi dali perspektivu u kritičnim trenucima doktorata kada se kraj činio nedostižnim.

Konačno, želim se zahvaliti roditeljima, bratu i supruzi na bezuvjetnoj podršci koju su mi pružali sve ovo vrijeme.

U Zagrebu, 21. srpnja 2022. godine.

ABSTRACT

This thesis addresses the problem of the visual localization of mobile agents in challenging scenarios. As challenging scenarios, we consider the environments populated by humans, i.e., the environments in which humans and robots coexist and cooperate. A good example of such an environment is an automated warehouse where humans and robots cooperate to increase the efficiency of the warehouse operation. The safety requirements for such coexistence are divided into three levels: (i) redirection of robots to avoid human-robot encounters, (ii) warning of humans about the environment and possible encounters with robots, and (iii) immediate shutdown of robots in close proximity to humans. The most important safety level is solved by setting up ranging sensors on all agents (humans and robots) in the warehouse and shutting down all robots whose distance to a human becomes too small. Less stringent safety requirements apply to the other two levels, and their implementation is allowed to be more complex. Since the cost of the solution is one of the criteria for choosing one solution over another, the focus is put on approaches that maximize the use of existing infrastructure in warehouses and require minimal installation time. The solution converged to a set of wearable sensors worn on the operator's Safety Vest which limits the processing power and power supply.

The implementation of the modified Semi-Global Matching (SGM) method for disparity computation was developed as part of the safety level that informs the operator of his environment and warns him of potential robot encounters. The computationally intensive steps of the original SGM are improved for an image sequence by reusing the existing disparity values from the previous steps. Assuming that the scene is constant in time, the disparity information between successive steps is transformed with visual odometry and fused with the new disparity measurements within the Kalman filter framework. The improvement in the complex steps of the method and the efficient implementation with the SIMD instruction set and multithreading showed the overall improvement of the proposed solution over the original SGM method. In addition, the Kalman filter framework enabled the detection of moving objects between two consecutive steps in the disparity images because moving objects do not follow the method's assumption of a static scene.

Locating all agents in the warehouse is a requirement for a safety level responsible for avoiding encounters based on the redirection of robots. The locations of the robots are already known to the warehouse management system, since tasks to carry racks are assigned based on their current location. The remaining element, localization of humans, requires a special approach since the conditions in the warehouse do not satisfy the static envi-

ronment assumption of all common localization approaches. Therefore, the implemented localization algorithm is based on two location cues: (i) visual odometry and (ii) detection of the existing warehouse's fiducial ground markers. The location cues are fused within the graph optimization process which provides a globally correct location estimate at a constant frequency. The proposed localization method is robust to visual aliasing and changing environment conditions in the warehouse, and requires only a lightweight map of ground marker placement. Furthermore, localization in visually challenging scenarios is further improved by augmenting the camera sensors with non-visual sensors and providing a simple odometry uncertainty model to provide information of visual odometry estimate quality.

KEYWORDS: semi-global matching, Kalman filter, moving objects detection, stereo camera, UWB sensors, wearable sensors fusion, warehouse localization, visual odometry, graph optimization-based localization

SAŽETAK

Robusna stereo vizualna lokalizacija mobilnih agenata u zahtjevnim scenarijima

U početku spominjani samo u znanstveno fantastičnim romanima prve polovice 20. stoljeća, roboti su brzo ušli u stvarnost pojavom prvih digitalnih računala 1960-ih. Svoj početak su započeli automatizacijom industrijske proizvodnje prvim komercijalnim robotom UNIMATE [1] i ubrzo su postali vrlo popularni jer su ljudi prepoznali potencijal robota pri obavljanju zamornih poslova u kojima se skup pokreta cijelo vrijeme ponavlja. Od tih početaka pa do danas, roboti su proširili svoje područje djelovanja i danas se koriste u skoro svih sferama života, od istraživanja svemira, medicine, pakiranja lijekova, transporta, proizvodnje hrane pa do obavljanja kućanskih poslova. Područja primjene robota su raznovrsna, ali u svim područjima konačni cilj primjene robota je isti: poboljšati kvalitetu života ljudi prepuštajući robotima da obavljaju zamorne i opasne poslove. Iako su početni roboti morali biti ili upravljani daljinskim upravljačem ili unaprijed programirani za točno definiran skup radnji, u novije vrijeme se počela istraživati i proizvoditi nova vrsta robota - autonomni roboti. Autonomni robotski sustavi imaju mnoge prednosti u odnosu na tradicionalne jer oni mogu raditi bez izravnog ljudskog nadzora što takve sustave čini učinkovitijima. Također, prilagodba izgleda i ponašanja autonomnih robota za obavljanje čovjeku kompliciranih zadataka, čini ih idealnim rješenjem i time pokazuje veliki potencijal područja autonomne robotike.

Moderni sustavi lanca opskrbe su zbog globalizacije postali vrlo složeni. Razvoj komunikacijskih tehnologija i transportnih sustava omogućili su da se materijali iz jednog dijela svijeta koriste za izradu proizvoda u drugom dijelu svijeta, i na kraju, da se proizvod prodaje na nekom trećem mjestu. U sustavu lanca opskrbe, jedan od važnijih elemenata čine skladišta koja čuvaju robu sve dok ona nije potrebna negdje dalje u lancu. Razvoj internetske kupovine popraćen je rastom skladišnih prostora u svim dijelovima svijeta i očekivan je nastavak tog pozitivnog trenda u idućim godinama [2]. Takav pozitivan trend i prisutnost velikih skladišta u svim dijelovima svijeta čine to područje zanimljivim i sustavima automatizacije, koji prilagodbom za skladišne potrebe povećavaju njihovu učinkovitost. U novije vrijeme inovacije iz područja robotike i IT industrije uspješno su integrirane u rad skladišta, te je predviđeno više nego dvostruko povećanje tržišta za robote u skladištima [3]. Praćenje robe u skladištu olakšano se prati nosivim senzorima, pametnim telefonima, bar kôdovima i RFID oznakama, dok je kretanje robe unutar skladišta također olakšano

sustavima automatizacije u kojima roboti autonomno prenose teške police i dostavljaju ih na odredišnu točku. Učinkovitost automatiziranih sustava se očituje i u tome što se takvi sustavi ne umaraju i njihova koncentracija ne opada s vremenom. Kapacitet skladišta opremljenih takvim sustavom je povećan jer se police više ne moraju oblikovati prema ljudskim standardima, već je moguće optimirati oblik polica i oblik robota kako bi se postigao maksimalan skladišni kapacitet. Takvi sustavi automatizacije udaljili su čovjeka od poslova u skladištu koji često uključuju zamoran, monoton i težak fizički rad u kojem su moguće i ozljede. Međutim, dok automatizirani dijelovi skladišta obično obavljaju poslove dizanja, prenošenja i sortiranja robe, ljudi obavljaju poslove kao što su primanje robe iz sustava, te poslove poput nadzora i održavanja automatiziranih dijelova.

Primjer suradnje automatiziranog sustava s ljudskim operaterima je CarryPick sustav kompanije Swisslog, [4] U CarryPick sustavu, skupina autonomnih mobilnih robota dostavlja police na sabirnu stanicu, gdje operater uzima željeni proizvod s police. Tako je ljudskom operateru olakšan posao jer ne mora osobno hodati do police s proizvodom koji želi preuzeti. Osim smanjenja potrošnje energije operatera, njegova pažnja može biti potpuno usmjerena na sabirnu stanicu i time se minimizira mogućnost grešaka. Mobilni roboti se lokaliziraju u skladištu i šalju podatke o svom trenutnom položaju sustavu za upravljanje skladištem (engl. *Warehouse Management System* - WMS). WMS maksimizira učinkovitost rada skladišta dodjeljujući zadatke robotima ovisno o njihovim trenutnim pozicijama i potrebama za nošenjem polica. Sigurnost operatera u takvom skladištu je zajamčena odvajanjem prostora na dio koji sadrži police i pripada radnom prostoru robota, te sabirne stanice koje pripadaju radnom prostoru operatera. Ta dva radna prostora su fizički odvojena sigurnosnom barijerom i prilikom prolaska kroz sigurnosnu barijeru odašilje se signal sustavu za upravljanje skladištem. Iako operateri po potrebi smiju ući u radno područje robota, sigurnosni zahtjevi nalažu za sustav upravljanja skladištem zaustavi sve robote po primitku informacije o prijelazu sigurnosne barijere, kako bi se izbjegla mogućnost nezgode. Roboti ostaju nepomični sve dok se operater ne vrati iz radnog područja robota i obavijesti sustav o svom povratku. Takav sigurnosni sustav je učinkovit što se tiče zaštite operatera, no kod velikih skladišta, učestali ulasci operatera mogli bi smanjiti učinkovitost rada skladišta što posljedično dovodi do oportunitetnih troškova. S druge strane, modifikacijom sustava u kojem se zaustavljaju samo roboti u blizini operatera, moguće je zadržati istu učinkovitost zaštite operatera, ali ujedno i smanjiti utjecaj operatera na učinkovitost rada skladišta.

Cilj projekta SafeLog [5] iz programa Europske unije Obzor 2020 je razviti sustav za sigurnu interakciju čovjeka i robota u automatiziranim skladištima kao što je CarryPick sustav. Sigurnosni koncept temelji se na tri razine sigurnosti:

- Sigurnosna razina A zaustavlja sve mobilne robote koji se približe operateru
- Razina sigurnosti B obavještava operatera o približavanju robota koji još ne predstavljaju prijetnju njegovoj sigurnosti
- Sigurnosna razina C planira putanje operatera i mobilnih robota kako bi se minimizirali njihovi bliski susreti

Koncept se oslanja na sigurnosni prsluk koji je opremljen senzorima, napajanjem i procesorskom jedinicom, te omogućuje komunikaciju s ostalim elementima sustava u skladištu.

Sigurnosni prsluk je obavezan odjevni predmet prilikom ulaska u skladište za svo osoblje. Senzori na sigurnosnom prsluku su odabrani tako da mogu ostvariti zadatke definirane razinama sigurnosti. Točnije, senzori (i) mjere udaljenost između čovjeka i mobilnih robota, (ii) prate robote u okolini i (iii) lokaliziraju čovjeka u skladištu. Mjerenja udaljenosti između sigurnosnog prsluka i svakog mobilnog robota koriste se kako bi se osigurala sigurnost operatera tijekom obavljanja zadatka. Sigurnost je osigurana trenutnim zaustavljanjem robota koji uđu u sigurnosnu zonu operatera. Nadalje, sustav za praćenje na sigurnosnom prsluku detektira kretanje robota i upozorava operatera na robote koji mu se mogu približiti. Osim toga, lokacija operatera se šalje u sustav upravljanja robotima (engl. *Fleet Management System* - FMS), koji u skladu s lokacijama operatera mijenja trenutne putanje mobilnih robota. Putanje se aktivno mijenjaju ako operater ne slijedi očekivani put, stoga je važno redovito ažurirati lokaciju operatera. Koncept sigurnosti ima za cilj omogućiti ljudskim operaterima i robotima da istovremeno izvršavaju zadatke u skladištu, ostavljajući minimalne tragove na učinkovitost skladišta.

Disertacija je organizirana u šest poglavlja. Prvo poglavlje je uvodno poglavlje i u njemu se opisuje problem, znanstvene doprinose i daje se kratak sadržaj disertacije po preostalim poglavljima. U drugom poglavlju se daje teorijska podloga znanstvenih doprinosa i opisuju se skupovi podataka korišteni za evaluaciju razvijenih metoda. Treće, četvrto i peto poglavlje predstavljaju po jedan od znanstvenih doprinosa, a šesto poglavlje zaključuje disertaciju i predstavlja ideje o daljnjem razvoju predstavljenih metoda. U nastavku slijedi kratki opis ostvarenih znanstvenih doprinosa.

#1 *Računski učinkovita metoda za estimaciju disparitеta koja koristi na stereo vizualnoj odometriji zasnovano smanjivanje prostora pretraživanja disparitеta.*

Poznavanje strukture scene od velike je važnosti robotu jer na temelju te informacije može rasuđivati o prostoru slobodnom za kretanje i prepoznavati elemente scene. Jedan od često korištenih senzora u robotici za određivanje strukture scene je stereo kamera. Na temelju slika lateralno postavljenih kamera s fiksnim odmakom struktura scene se određuje preko disparitеta. Razvijeno je više pristupa za problem računanja disparitеta, a među najčešće korištenima je metoda poluglobalnog podudaranja (engl. *Semi-Global Matching* - SGM). Predstavljena metoda je inspirirana projektom SafeLog u kojem je naglasak stavljen pristupe koji se izvršavaju u stvarnom vremenu na platformi ograničenih resursa. U takvom scenariju potrebno je ponovno iskoristiti što više postojećih informacija koje mogu pomoći pri estimaciji tražene vrijednosti.

Problem svih metoda za računanje disparitеta je određivanje dijelova slika koje su projekcija istog elementa scene (engl. *correspondence problem*). Uz pretpostavku da se scena približno slično preslikala u obje slike, očekivano je da dijelovi slike lijeve i desne kamere koji predstavljaju projekciju istog elementa scene najviše sliče jedan drugom. U standardnoj metodi SGM vremenski najzahtjevniji koraci su oni koji određuju sličnost uzoraka lijeve i desne kamere, te oni koji na temelju sličnosti uzoraka i pretpostavke o kontinuitetu scene zaključuju o vrijednosti disparitеta. Jedan od ulaznih parametara metode je prostor pretrage unutar kojeg se uzimaju uzorci za provjeru sličnosti. Što je prostor pretrage veći, to je složenije izvođenje metode.

Unutar doprinosa, razvijena je metoda koja učinkovito računa disparitet na slijedu slika stereo kamere. Učinkovitost se temelji na (i) reduciranoj kompleksnosti metode smanjivanjem prostora pretrage dispariteta, te (ii) implementaciji pomoću instrukcijskog seta procesora "jednostruka naredba, višestruki podaci" (engl. Single Instruction, Multiple Data - SIMD). Pri smanjenju prostora pretrage korištena je informacija o izgledu scene iz prethodnog koraka. Uz pretpostavku da je scena statična, te da u slijedu slika svo gibanje u slikama dolazi od promjene položaja stereo kamere, mogu se pretpostaviti vrijednosti dispariteta u sljedećem trenutku. Preslikavanje vrijednosti dispariteta između trenutaka temeljeno je na metodi vizualne odometrije koja određuje pomak kamere između dva uzastopna vremenska trenutka. Pretraživanje nove vrijednosti dispariteta time je suženo na područje oko pretpostavljene vrijednosti dispariteta, te je širina tog područja definirana nesigurnošću pretpostavke. Vremensko praćenje dispariteta, nesigurnosti dispariteta, te estimacija novog dispariteta na temelju smanjenog područja pretrage implementirani su u okviru Kalmanovog filtra. Također, predstavljen je način mjerenja procesnog i mjernog šuma u modelima Kalmanovog filtra.

Pretpostavka predstavljene metode o statičnoj sceni nije uvijek ispunjena, te u slučaju objekata koji se gibaju neovisno od kamere dolazi do nepoklapanja između pretpostavljene i izmjerene vrijednosti dispariteta. Uočavanjem veće količine neočekivanih vrijednosti u određenom dijelu slike, moguće je detektirati neovisno gibajući objekt u sceni. U sklopu doprinosa predstavljen je pristup detekciji gibajućih objekata.

Predstavljena metoda za učinkovitu estimaciju dispariteta evaluirana je na skupu podataka KITTI [6] i uspoređena je sa standardnom metodom SGM, te metodom dubokog učenja LEAStereo koja je jedna od vodećih metoda na KITTI ljestvici za računanje dispariteta. Detekcija gibajućih objekata evaluirana je na KITTI MOD skupu podataka.

#2 *Metoda stereo vizualne lokalizacije autonomnih agenata u robotiziranim skladištima zasnovana na nosivim senzorima.*

Aktualna lokacija ljudskog operatera u skladištu preduvjet je za povećanje učinkovitosti rada skladišta temeljen na planiranju putanja robota. Skladište u kojem autonomni roboti prenose police specifičan je slučaj lokalizacije jer okolina mijenja svoj izgleda u vremenu. Većina postojećih metoda za lokalizaciju temeljena je na nepromjenjivom izgledu okoline i njihova primjena bi imala ograničen uspjeh. Također, neke od metoda osim lokalizacije postepeno grade kartu okoline koju potom koriste za lokalizaciju. Osim pretpostavke o nepromjenjivosti, karta takvih pristupa može postati kompleksna u slučaju većih skladišta. Uz ograničene procesne resurse kakvi se očekuju od računala pričvršćenog za sigurnosni prsluk, za lokalizaciju je potrebno koristiti kartu čija se složenost dobro skalira s veličinom skladišta. U skladu s generalnom idejom projekta, metoda za lokalizaciju treba maksimalno iskoristiti postojeće informacije iz skladišta i minimizirati korištenje informacija koje zahtijevaju posebne promjene u skladištu, te time povećavaju vrijeme i cijenu instalacije sigurnosnog koncepta.

Doprinos predstavlja metodu usmjerenu na lokalizaciju ljudskih operatera u okruženju automatiziranih skladišta u stvarnom vremenu, gdje je cijeli sustav lokalizacije baziran na elementima nošenim na sigurnosnom prsluku. Metoda se temelji na fuziji informacija

lokacije dobivenih iz vizualnih senzora, stereo kamere i monokularne kamere, pomoću metoda vizualne odometrije i detekcije pouzdanih podnih markera. Podni markeri se već nalaze u automatiziranim skladištima gdje ih roboti koriste za određivanje svog položaja. Detekcijom podnih markera moguće je odrediti položaj operatera unutar skladišta. Takav izvor lokacije koristi postojeću kartu markera, te je karta markera memorijski nezahtjevna i nepromjenjiva u vremenu. Međutim, vremenska nepredvidivost detekcija ne zadovoljava zahtjeve sigurnosnog koncepta, pa se informacije o lokaciji temeljene na detekciji markera spajaju s informacijama vizualne odometrije koja estimira relativan pomak sigurnosnog prsluka. Spajanjem ta dva informacijska izvora lokacije u okviru optimizacije grafa dobiva se aktualan globalni položaj operatera u skladištu.

Implementacija predstavljene metode evaluirana je na vlastitim, javno dostupnim, skupovima podataka i uspoređena je s implementacijom ORB-SLAM2 metode koja predstavlja stanje tehnike za simultanu lokalizaciju i kartiranje (engl. *Simultaneous Localization and Mapping* - SLAM).

#3 *Postupak za poboljšanje robusnosti stereo vizualne lokalizacije autonomnih agenata u zahtjevnim scenarijima.*

Specifičnost izgleda unutrašnjosti skladišta i svjetlosnih uvjeta nije uvijek najpovoljnija za lokalizaciju pomoću vizualnih senzora. Ponekad je praćenje vizualnih značajki otežano zbog lošeg osvjetljenja, slabe teksture u sceni ili zbog ograničenog vidnog polja kamere, što smanjuje kvalitetu estimirane lokacije operatera. Pri lokalizaciji operatera pomoću optimizacije grafa, vrlo je bitno znati kvalitetu informacije elemenata u grafu, odnosno prepoznati pouzdanost izvora lokacije. Na taj način će se pri optimizaciji više izmijeniti elementi grafa koji su manje pouzdani. Također, u slučajevima smanjene kvalitete vizualnih izvora lokacije, veća se robusnost može postići dodavanjem izvora lokacije koji se ne temelje na vizualnim podražajima.

U sklopu doprinosa je predstavljena izmijenjena metoda lokalizacije operatera u skladišnim okruženjima opisana u prethodnom doprinosu. Izmjene su usmjerene na poboljšanje rada lokalizacije u slučajevima kada je kvaliteta vizualnih izvora lokacije narušena. Prva izmjena je uvođenje nevizualnog izvora lokacije koji se temelji na ultra širokopojasnim senzorima (engl. *Ultra Wide Band* - UWB). UWB senzori kroz komunikaciju vrlo precizno određuju međusobnu udaljenost te se u sigurnosnom konceptu skladišta koriste za određivanje udaljenosti operatera i robota. S obzirom na to da sustav upravljanja skladištem zna pozicije svih robota, a UWB senzori mjere udaljenost između robota i sigurnosnog prsluka, moguće je postupkom trilateracije odrediti poziciju operatera u skladištu. Pozicija dobivena trilateracijom predstavlja novi, nevizualni izvor lokacije koji je dodan ostalim izvorima u optimizacijskom grafu. Za lokalizaciju korištenjem UWB senzora nije potrebna posebna prilagodba hardvera ili komunikacijskih kanala jer se postojeće informacije već koriste za drugu svrhu. Druga izmjena je prilagodba težina bridova u optimizacijskom grafu. U optimizacijskom grafu dva čvora koja predstavljaju stanje povezana su bridom koji predstavlja estimaciju relativnog pomaka dobivenog vizualnom odometrijom. Izvorno predstavljena metoda otežavala je jednakom težinom sve bridove dobivene vizualnom odometrijom. Naknadno je uočeno da takav način otežavanja može dovesti do pogrešnih estimacija

lokacije operatera, te izmijenjena metoda otežava bridove na temelju procijenjene kvalitete estimacije vizualne odometrije. Za određivanje kvalitete estimacije osmišljen je jednostavan model greške vizualne odometrije, te se relativni pomaci s manjom procijenjenom greškom otežavaju većom težinom.

Predstavljene izmjene su evaluirane na vlastitim skupovima podataka, snimanih u okolini koja je vizualno slična skladištu, a sastoje od snimaka s mjerenjima UWB senzora u različitim vizualnim uvjetima. Izmijenjena metoda je uspoređena s originalnom metodom i ORB-SLAM2 metodom.

KLJUČNE RIJEČI: poluglobalno podudaranje, Kalmanov filter, detekcija gibajućih objekata, stereo kamera, UWB senzori, stapanje informacija s nosivih senzora, lokalizacija u skladišnim okruženjima, vizualna odometrija, lokalizacija temeljena na optimizaciji grafa

CONTENTS

1 INTRODUCTION 1
- 1.1 Motivation and problem statement 1
- 1.2 Original contributions 3
- 1.3 Outline of the thesis 4

2 BACKGROUND 7
- 2.1 Visual sensors 7
  - 2.1.1 Camera model 7
  - 2.1.2 Visual depth estimation 10
  - 2.1.3 Stereo cameras 11
- 2.2 Visual localization 16
  - 2.2.1 Localization problem 17
  - 2.2.2 Localization with visual sensors 19
- 2.3 Datasets for evaluation 22
  - 2.3.1 KITTI and KITTI MOD datasets 22
  - 2.3.2 Self recorded datasets 23
  - 2.3.3 Evaluation metrics 28

3 DISPARITY ESTIMATION USING STEREO VISUAL ODOMETRY 30
- 3.1 Classic SGM 31
  - 3.1.1 Matching cost computation 31
  - 3.1.2 Cost aggregation 32
  - 3.1.3 Disparity computation and refinement 33
- 3.2 Reduced disparity-search space SGM 33
  - 3.2.1 Kalman filter 34
  - 3.2.2 Reduced space SGM 36
  - 3.2.3 Disparity estimation with the Kalman filter 37
- 3.3 Estimation of process and measurement noise 40
- 3.4 Moving object detection with a dynamic stereo camera 40
- 3.5 Evaluation 43
  - 3.5.1 Comparison of proposed and OpenCV SGM implementation 43
  - 3.5.2 Comparison of the proposed and learing-based approach 47
  - 3.5.3 Moving objects detection test 47
- 3.6 Summary 48

4 STEREO VISUAL LOCALIZATION OF AUTONOMOUS AGENTS IN ROBOTIZED WAREHOUSES 51
4.1 Motivation 52
4.2 Globally correct visual localization with fiducial markers 54
4.3 Relative localization with visual odometry 56
4.4 Fusion of global and relative location cues in graph optimization framework 59
4.4.1 Graph optimization 59
4.4.2 Fusion of location cues 61
4.5 Evaluation 63
4.5.1 Dortmund dataset 64
4.5.2 Augsburg dataset 68
4.6 Summary 74

5 ROBUSTNESS IMPROVEMENT OF STEREO VISUAL LOCALIZATION USING UWB SENSORS 76
5.1 Motivation 76
5.2 Globally correct pose estimation with UWB sensors network 78
5.3 Variable odometry edge weights in optimization graph 80
5.3.1 Uncertainty due to traveled distance 81
5.3.2 Uncertainty due to lack of features 82
5.4 Evaluation 83
5.4.1 Dortmund and Augsburg datasets 84
5.4.2 Zagreb dataset 85
5.5 Summary 88

6 CONCLUSIONS AND OUTLOOK 90

BIBLIOGRAPHY 94

CURRICULUM VITAE 111

ŽIVOTOPIS 113

1

Introduction

MOTIVATION AND PROBLEM STATEMENT

Initially mentioned only in the science fiction novels of the first half of the 20th century, robots quickly found their way into the real world with the advent of digital computers in the 1960s. Their journey into the real world began with the first commercial robot, UNIMATE [1], in industrial manufacturing. Their popularity grew rapidly as people realized the potential of robots to relieve humans of repetitive and dangerous tasks. From then on, robots grew beyond the realm of industrial manufacturing and are now used in space exploration, medicine, transportation, food production, and many other fields. All of these applications have one goal: to improve the quality of life by letting robots do heavy and dangerous tasks. In the last century, robots were either remotely controlled or pre-programmed to perform repetitive motions, but in today's world, the focus is on autonomous robotic systems because they offer many advantages over traditional systems. They can operate without direct human supervision, leading to greater efficiency and effectiveness. Customization to a specific task makes them ideal for certain applications, and the field of autonomous robots shows great potential.

Today's supply chain systems have become very complex due to globalization. Improvements in communication and transportation systems have made it possible for materials from one part of the world to be used to manufacture products in another part of the world, only to be sold somewhere else. One of the most important elements of the supply chain is warehouses, where goods are stored until they are needed further down the chain. According to the IMARC Group [2], the global warehousing and storage market reached $451.9 billion in 2021 and is expected to reach $605.6 billion by 2027. Moreover, the high demand for warehouse space in developed and emerging markets is a result of the increasing popularity of online shopping. The growing potential of large and complex warehouses is fertile ground for automation systems that can support their efficiency. Innovations in robotics and IT have been successfully integrated into warehouse operations. The warehouse robotics market, estimated at $5.04 billion in 2021, is expected to more than double in value by 2027 [3]. Tracking of goods in warehouses is facilitated by wearable sensors, smartphones, barcodes, and RFID tags. In addition, the movement of goods within the warehouse is being improved by automation technologies such as goods-to-person (GTP), automated guided vehicles (AGVs), autonomous mobile robots (AMRs), and automated sortation systems. These technologies have moved people away from warehouse jobs that involve

tedious, repetitive, and hard physical labor with the risk of injury. Automation systems are highly effective because they do not tire and their concentration does not diminish over time. In addition, the automation systems can increase the capacity of the warehouse, such as GTP systems, since racks and other elements do not need to be shaped according to human standards. However, the automated warehouses usually perform lifting, picking and sorting tasks, while humans perform tasks such as receiving goods from the system, system monitoring and maintenance in cooperation with the automated system. An example of an automated storage system with human workers is the CarryPick system from Swisslog [4]. In the CarryPick system, the fleet of autonomous mobile robots delivers shelves to picking stations, and the human worker at the picking station picks a desired product from the shelf. The system shortens the path of the worker, who can focus on the picking station, minimizing the picking error and saving energy. The mobile robots of the CarryPick system are able to locate themselves in the warehouse and send their position data to the warehouse management system (WMS). The WMS maximizes warehouse efficiency by assigning tasks to the robots based on their position and the current layout of the racks. In such a system, the warehouse is divided into a work area for the robots, which covers the part with the racks, and a work area for the humans, which focuses on the picking stations. The two areas are separated by a security fence and an alarm signal is sent to the WMS when the fence is breached. Although humans can enter the area with the robots if needed, safety requirements dictate that the WMS stops all robots as soon as the alarm signal is sent. The robots do not move until the human returns back from the area with the robots and notifies the WMS. In large warehouses, such scenarios have a greater impact on warehouse efficiency, which in turn leads to opportunity costs. However, safety requirements can be met with much less impact on efficiency by stopping only the robots near the human.

The goal of the European Horizon 2020 project Safelog [5] is to develop a safety system for safe human-robot interaction in automated warehouses. The safety concept is based on three safety levels

- Safety Level A stops all mobile robots that come close to the human worker
- Safety Level B informs the worker of approaching robots that do not yet pose a threat to his safety
- Safety Level C plans the paths of human workers and mobile robots to minimize close encounters

The safety system relies on the Safety Vest, which is equipped with sensors, power supply and CPU and communicates with the rest of the system. The Safety Vest must be worn by all personnel entering the warehouse. The sensors on the Safety Vest must: (i) measure the distance between the human and the mobile robots, (ii) track the robots in the environment, and (iii) locate the human in the warehouse. The distance measurements between the Safety Vest and each mobile robot are used to ensure the safety of the human while performing his task. Safety is ensured by briefly stopping the robots that enter the human's safety zone (Safety Level A). The tracking system on the Safety Vest detects the robots' movements and warns the human of the robots that may approach him. In addition, the human's location is sent to the Fleet Management System (FMS) (Safety Level C), and the FMS changes

the current routes of the mobile robots accordingly. Routes are actively rescheduled if the human does not follow the expected path. Therefore, it is important to update the human's location regularly. The safety concept aims to allow humans and robots to coexist in the warehouse while leaving minimal traces in the warehouse efficiency.

ORIGINAL CONTRIBUTIONS

The original scientific contributions are closely related to the safety concept of the Safelog project. Namely, the Safelog project concept is based on wearable devices, which limits the size and weight of the sensors, power supply, and computing power. Problems in such a use case require approaches that take maximum advantage of the existing constraints in the system. The contributions focus on making inferences about the warehouse environment, by maximally exploiting all available information, which helps to reduce the computational complexity of the solution. The contributions with a brief explanation can be found in the sequel.

#1 A computationally efficient method for disparity estimation using stereo visual odometry based disparity search space reduction.

The developed method efficiently computes the disparity in a sequence of stereo camera images. The method focuses on simplifying the most computationally complex steps of the state-of-the-art disparity approach Semi-global matching (SGM). Assuming that the subsequent stereo pairs in the sequence share most of the same scene and that the scene does not change over time, the disparity from the previous step can be used to compute the new disparity. The disparity from the previous step is transformed with the information from the visual odometry computed on the same images. Instead of searching for disparity values for all pixels in the predefined range, the initial estimate is used as the central point of the disparity search space, whose width is defined by the uncertainty of the initial disparity estimate. The uncertainty of the estimate, the disparity transformation with visual odometry, and the new disparity measurement computed on the reduced search space are integrated into the recursive Kalman filter framework. The implementation showed improved performance in both accuracy and runtime over the original SGM method.

#2 A method for stereo visual localization of autonomous agents in robotized warehouses supported by other wearable sensors.

The contribution presents a method for localization of agents in robotized warehouses based solely on wearable visual sensors. Motivated by the idea of reusing environmental conditions, visual sensors were chosen because fiducial markers on the ground already exist for the localization of mobile robots. However, warehouse environments pose a difficult problem for visual localization due to lighting conditions, visual aliasing, and the dynamic arrangement of racks. The fiducial marker detection provides a globally correct Safety Vest pose, but due to the unpredictable movements of humans, pose estimates are occasional, which is inappropriate for FMS replanning. To improve the frequency of pose estimation, fiducial marker detection is combined with visual

odometry. Visual odometry provides a relative pose estimate at a constant frequency, but the uncertainty of the estimate is progressively increasing. The fusion of the two location cues, which is implemented in the graph optimization framework, results in a globally correct pose estimate at a constant frequency and thus satisfies the requirements of the FMS. Visual localization is robust to visual aliasing and changes in rack layout because it uses a lightweight map of fiducial markers. The implemented method is tested on self-recorded datasets and compared with the state-of-the-art SLAM method.

#3 A procedure for robustness improvement of stereo visual localization of autonomous agents in challenging scenarios.

The contribution presents an improvement of localization based on visual sensors fusion. The challenging lighting conditions in the warehouse can affect the performance of visual sensors and thus the location estimation. The robustness of visual localization under such conditions is improved by adding the location cue from a sensor in the non-visual domain. Since the safety of the people wearing the Safety Vest is ensured based on the non-visual range measurements that stop robots in their vicinity, and the WMS knows the poses of all mobile robots, the idea is to reuse this information to compute the pose of the Safety Vest. Following the idea of maximally reusing the available information due to the limited resources, the non-visual location cue is included in the fusion framework without introducing any new hardware or software components. In addition, the robustness of visual localization is improved by a visual odometry error model. Originally, visual sensor fusion was equally trusting to all odometry estimates in the graph optimization framework. With the proposed odometry error model, the quality of visual odometry can be estimated and when applied in graph optimization, the optimization will trust less to the relative transformations with the increased error estimate.

OUTLINE OF THE THESIS

The thesis is divided into six chapters, beginning with this chapter, the introductory chapter. The following chapter provides an overview of the theory underlying the presented contributions and describes the datasets used to evaluate the implemented methods. Each of Chapters 3-5 presents one of the scientific contributions. These chapters begin with the introduction or motivation for the problem, followed by the theoretical background of the solutions. After the evaluation on datasets introduced in the second chapter, the chapters end with a summary that discusses the results of the evaluation and concludes the presented work. The sequel provides a brief overview of the content of each chapter.

Ch 2 This chapter is divided into two parts: the first part presents the theoretical background of the contributions in the thesis, and the second part describes the datasets used for the evaluation of implemented methods. The first part starts with the mathematical models of the pinhole monocular and stereo cameras, which are the basis for all the following concepts based on visual sensors. Following the stereo camera model, the idea of the disparity in stereo camera images and its usefulness in environment

inference is described. The dense disparity estimation approach is presented since it is a subject of research in the first contribution. Then, the problem of localization is presented by stating the problem formulation and giving an overview of the existing approaches. Special attention is given to visual localization since the second and the third contribution are engaged with the localization based on visual sensors. The second part of the chapter describes the evaluation datasets. The first dataset, the KITTI dataset, is commonly used in the robotics community since it has evaluation benchmarks for several well-known robotics problems. An extension of the KITTI dataset, KITTI MOD, used for the evaluation of the moving objects detection is also introduced. The other three datasets are self-made for the evaluation purposes of the method presented in Chapter 4 and Chapter 5.

Ch 3 This chapter begins with the introduction of the semi-global matching (SGM) method for disparity computation. Each of the SGM steps is described, emphasizing the computationally complex parts. The description of the SGM method is followed by the proposal on how to improve the complexity of the original implementation when the disparity is computed for a sequence of images. The proposal is based on the assumptions that the image sequences have a certain amount of the common scene and that objects in the scene are static. After introducing the workflow of the Kalman filter, the idea of exploiting the disparity information from the previous image pair to reduce the disparity computation complexity is presented through the steps of the Kalman filter. However, a fusion of information through the Kalman filter requires setting two values, the process and the measurement noise. Thus, the method is presented on how to determine the process and measurement noise using the data from the KITTI dataset. Afterwards the idea of detecting the independently moving objects on the image sequence from the moving stereo camera is proposed. Since all the above-mentioned work is based on the assumption that the environment is static, the moving objects are recognized as clusters of outliers between the expected and measured disparity. The proposed ideas, the reduced search space SGM and moving objects detection, are evaluated on the KITTI and KITTI MOD datasets.

Ch 4 The chapter begins with the introduction to the Safelog project and the safety concept developed for the human-robot coexistence in warehouse environments. The chapter deals with one of the project's components related to the localization of humans based on wearable visual sensors. After introducing the problem and its constraints, the building blocks of the localization method are described. The localization is based on the fusion of two visual location cues. The first one comes from the detection of the fiducial markers, installed on the warehouse floor and used for the localization of mobile robots. This location cue gives a globally correct pose estimate of the human wearing the visual sensors, but the infrequent nature of this location cue makes it insufficient for the safety concept use case. Thus, the visual odometry is introduced as a second location cue which gives a relative pose estimate at a constant frequency. Afterward, the graph optimization is described, as it is used for the fusion of information from the two location cues. The evaluation is performed on the self-recorded Dortmund and Augsburg datasets involving special scenarios like the kidnapped-human

scenario and the non-static environment scenario.

Ch 5 This chapter presents the extension of the method proposed in the previous chapter. In the beginning, the motivation is given by emphasizing the specific visual conditions in the warehouses and the need for a modification of the original localization method which will make the localization more robust in the challenging conditions. The two modifications are introduced: 1) the non-visual location cue is added to the graph optimization framework, and 2) the information of visual odometry quality is incorporated into the optimization graph. Following the motivation, the information flow in the safety concept and the warehouse is explained. Based on this information flow it is shown how the new, non-visual, location cue can be introduced using only already existing information in the warehouse. After, the simple model is introduced which estimates the amount of odometry error based on two recognized error sources: 1) the error coming from the magnitude of motion, and 2) the error caused by the low-textured scenes. The two introduced modifications are incorporated into the graph optimization framework and evaluated on the self-recorded datasets. Due to specific requirements of the newly introduced location cue, which do exist in the warehouse, but are not simulated in the Augsburg and Dortmund datasets, this modification is evaluated on the Zagreb dataset.

Ch 6 This chapter concludes the thesis by giving a summary of the presented contributions and an outlook on potential future improvements to the warehouse safety concept.

2

Background

This chapter begins with an introduction to the very popular sensors in robots, visual sensors. We focus specifically on visual cameras and describe the mathematical model of a pinhole camera with the radial-tangential lens distortion model. The overview of visual depth estimation approaches is given and the multiview approach with the stereo camera is described in more details. After depth estimation with a stereo camera via disparity, we give an overview of classic and modern approaches to disparity estimation. Later, we introduce the robot localization problem and present the elements that are important for visual localization methods. In the last part of the chapter, we describe datasets that were used to evaluate the method, presented in Chapters 3-5.

VISUAL SENSORS

Nowadays, there are different types of visual sensors, from standard cameras, to RGB-D cameras specialized to capture the depth information along with the image, to event cameras capable of capturing a scene at high speed and in high dynamic range by providing only the information about the difference of pixel intensities. In this chapter, we focus on standard visual cameras, which are used in the following chapters.

Camera model

A widely used and simplest camera model is the pinhole camera model. The pinhole camera model uses a central projection, popularly illustrated as a box with a small hole on one side and an image of the scene on the other side. Light rays pass through the hole, the *optical center*, and form a projection of the scene on the other side of the box, the *image plane*. For simplicity, a slightly reordered situation is shown in Fig. 2.1 where the image plane is placed between a scene's 3D point and the optical center C. A *focal length* f is the distance between the optical center C and a *principal point* $\mathbf{p}$, which is an intersection of the image plane and a *principal axis*.

As for the coordinate system of the projection, the projected point can be expressed in the *camera coordinate system* and the *image coordinate frame*. For some applications, it is convenient to express the projection in the camera coordinate frame, which is unaware of the intrinsic camera parameters, and the projection is done on the *normalized image plane* with focal length $f = 1$. In Fig. 2.1, the camera coordinate frame is defined with x, y, and

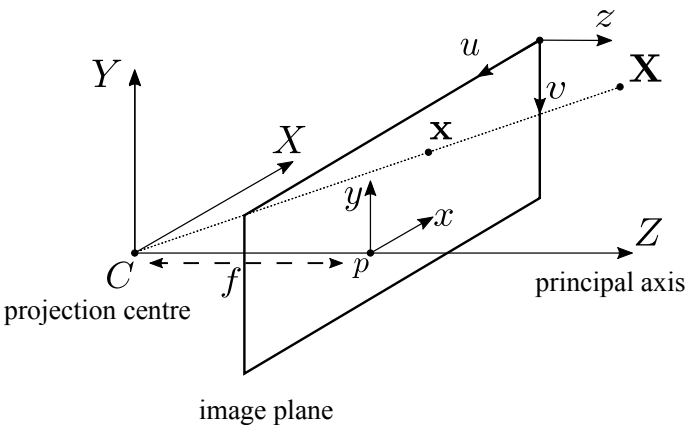


Figure 2.1: Pinhole camera model. A light beam coming from a 3D point **X** goes through the optical center C and intersects an image plane in point **x**.

the principal axis Z, coinciding with the image plane for simplicity (not true in the general case). However, the projected coordinates are usually expressed in the image coordinate frame, marked with u, v, and z axes in Fig. 2.1. This coordinate frame is located in the upper left corner of the image sensor. To express the projection in the image coordinate frame, one requires the parameters of the camera: the focal length f and the principal point $\mathbf{p} = (p_u, p_v)^T$.

The relationship between the 3D point $\mathbf{X}_W = (X_W, Y_W, Z_W)^T$ in the world coordinate frame W and its projection $\mathbf{x} = (u, v)^T$ onto the image plane is given by Equation (2.1).

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & p_u & 0 \\ 0 & f & p_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_C^W & t_C^W \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.1}$$

where $\mathbf{X}_W$ and $\mathbf{x}$ are expressed in homogeneous coordinates. The second right-hand side matrix transforms the 3D point from the world coordinate frame W into the camera coordinate frame C. Equation (2.1) is often written in the following forms

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R_C^W | t_C^W] \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.2}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.3}$$

where the matrix K is called the *camera calibration matrix* because it contains the intrinsic parameters of the camera, and the matrix P is called the *camera projection matrix* because it contains both intrinsic and extrinsic parameters of the projection.

Charged Coupled Device (CCD) cameras can have non-square pixels, so the first two diagonal elements of the camera calibration matrix in Eq. (2.3) are assigned additional

parameters:

$$K = \begin{bmatrix} f\alpha_u & 0 & p_u & 0 \\ 0 & f\alpha_v & p_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.4}$$

common notation merges the focal length with scaling factors (α_u, α_v) $f_u = f\alpha_u$ and $f_v = f\alpha_v$.

LENS DISTORTION. The ideal pinhole model is a simple approximation of the real camera model. The cameras replace a small hole of the pinhole model with a lens. The pinhole model approximation of the lens works well at the center of the lens, but farther away, nonlinear artifacts known as *lens distortion* occur. The two most popular lens distortion models are *Brown-Conrady* [7] or *radial-tangential distortion* (radtan) and *Kannala-Brandt* [8] or *equidistant distortion*. In the sequel, we give a brief description of the Brown-Conrady distortion model as it was used to model distortion in Chapter 4 and Chapter 5.

We begin with the ideal pinhole projection model from Eq. (2.1). The distortion model first projects the 3D point in the world coordinate frame $\mathbf{X}_W = (X_W, Y_W, Z_W)^T$ onto the normalized image plane in the camera coordinate frame. The transformation matrix $[R_C^W | t_C^W]$ transforms the 3D point $\mathbf{X}_W$ from the world coordinate frame to the camera coordinate frame ($\mathbf{X}_C = (X_C, Y_C, Z_C)^T$).

$$w' \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_C^W & t_C^W \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \tag{2.5}$$

The distortion model is applied on the normalized image plane projection $(u', v')^T = (X_C/Z_C, Y_C/Z_C)^T$.

The Brown-Conrady model is a superposition of the two image distortion sources: *radial distortion* and *tangential distortion*. Radial distortion is caused by the spherical shape of the lens and is more apparent in areas of the image farther away from the principal point. It is modeled with the following equations

$$u_d = u'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \ldots + k_n r^{2n}) \tag{2.6}$$

$$v_d = v'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \ldots + k_n r^{2n}) \tag{2.7}$$

where $r = \sqrt{u'^2 + v'^2}$. In practice, most of the distortion is contained in the first two to three elements of the sum while the rest is ignored.

The second source of distortion, tangential distortion, arises from the misalignment of the optical axis with the projection plane, and the distorted point (u_d, v_d) is described by the following equations:

$$u_d = u'(2p_1 u'v' + p_2(r^2 + 2u'^2)) \tag{2.8}$$

$$v_d = v'(p_1(r^2 + 2v'^2) + 2p_2 u'v') \tag{2.9}$$

Once we have the distorted point in the normalized image plane, it is transformed to the sensor plane in the image coordinate system

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & p_u \\ 0 & f_v & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} \tag{2.10}$$

where (p_u, p_v) are image coordinates of the principal point.

CAMERA CALIBRATION. Camera model parameters are divided into *intrinsic* and *extrinsic* parameters, and are determined by the calibration process. The intrinsic parameters describe the lens distortion and the projection model, while the extrinsic parameters describe the pose of the camera in space. The standard calibration procedure requires several images of a calibration target taken in different poses, Fig. 2.2. Since the model of the target is known, the calibration process attempts to mimic the projection and aims to minimize the distances between the real and mimicked projections of the target by tuning the model parameters. The calibration process knows the layout and metric dimensions of the target and can thus determine the relationship between the pixel scale and the real scale.

The commonly used Zhang's method [9] takes a series of images of the planar target taken from different perspectives. In the case of a checkerboard target, the intersections of two black squares and two white squares represent the set of salient points with known coordinates in the target's coordinate frame. Zhang's calibration method assumes that each image has its own coordinate frame placed on the calibration target, eliminating all Z coordinates of the salient points used in the calibration process. This simplifies the problem to the estimation of homography, which is solved using the direct linear transform (DLT). Once the homography is known, it provides the initial estimate of the intrinsic camera parameters. These parameters are further improved by nonlinear refinement by minimizing the projection error. A detailed derivation and pseudocodes for implementation can be found in [10] and the open source implementation is available in the OpenCV library [11].

Visual depth estimation

The projective property of cameras prevents them from making inferences about the depth of the 3D world. However, the depth of the scene is valuable information and different approaches have been developed to enable depth inference with visual sensors. Image-based depth perception has been actively studied in image processing and robotics community since digital cameras became popular. The methods developed can be classified as *active* and *passive*. In the active methods, light is emitted into the scene and depth is inferred from measurements of the reflected light. Examples of active methods are *Shape-from-Shading* [12], or *Active Stereo* [13] which is commonly used in RGB-D cameras such as Microsoft Kinect [14]. Passive methods rely solely on light coming from the scene, and depth inference is based on multi-view triangulation. When the multi-view images have unknown geometric relationships, both depth and camera poses are computed using the *Structure-from-Motion* method (SfM) [15]. On the other hand, a simpler approaches are used to compute depth when the geometric relationship is known. An example of cameras with known geometric relationship is a stereo camera.

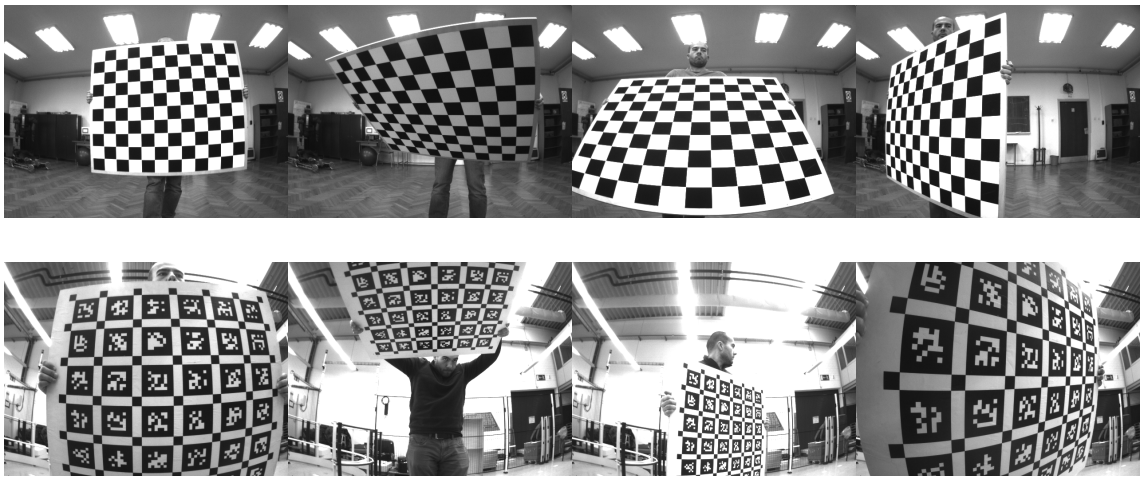


Figure 2.2: Calibration targets, checkerboard (upper) and aprilgrid (lower), recorded in several poses during the camera calibration process.

Stereo cameras

The *stereo camera* consists of two camera sensors that share a significant area of the overlapping scene. In the standard stereo model configuration, there is only a lateral displacement between the cameras. Figure 2.3 shows a pair of ideal pinhole cameras. The 3D points X_1-X_5 are all projected to the same point u_l in the image plane of the left camera, but they are also projected to 5 different points u_{r1}-u_{r5} in the image plane of the right camera. The difference in coordinates of u_l and u_{r1}-u_{r5} along with the known camera pose difference can be used to triangulate the relative positions of points X_1-X_5, respectively. The pose difference of the cameras in a stereo pair is constant and its parameters are determined by the extrinsic calibration of the cameras.

Usually, points u_l, and u_{r1}-u_{r5} do not stand out as clearly from the rest of the image as in Fig. 2.3 and matching points must be found. The *epipolar constraint* significantly improves the complexity of the matching process. It results from the geometry of the stereo camera, as shown in Fig. 2.3, and limits the search area for the corresponding point of one camera image, e.g. u_l, along one line in the other image. This line, called an *epipolar line*, connects a projection of a 3D point and the optical center of the other camera, called an *epipole*.

The *fundamental matrix* F is the epipolar constraint in algebraic form. This 3×3 matrix maps a point in homogeneous coordinates from one camera image to a line in homogeneous coordinates in the second camera image, Eq. (2.11).

$$l' = Fx \tag{2.11}$$

where x is a point in the first image and l' is a line in the second image.

Since for a point x on a line l the following is valid $x^T l = 0$, we can rewrite equation (2.11) to a form that connects matching points in the images, Eq. (2.12).

$$0 = x'^T F x \tag{2.12}$$

Image processing, especially for real-time applications, has high demands on computation time. Therefore, in applications that require many matching points, the images are

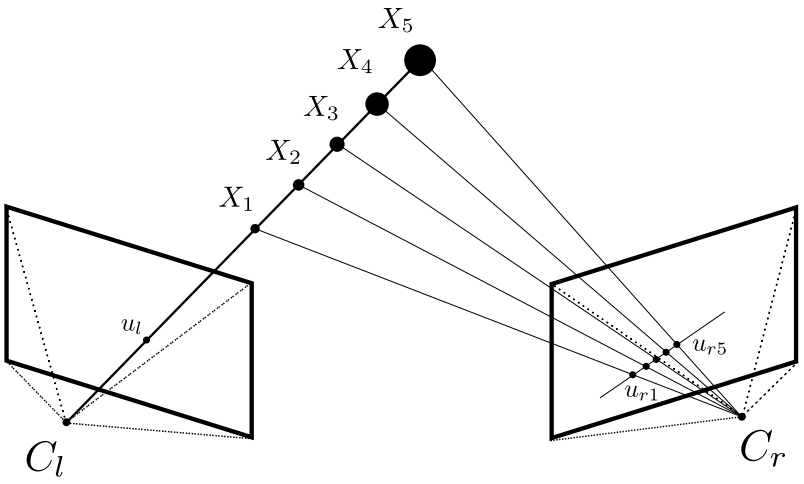


Figure 2.3: A stereo camera model: 3D points X_1-X_5 project to the same point u_l in the left image plane, but in the right image plane they project to 5 different points, u_{r1} - u_{r5}.

rectified in a preprocessing step. When stereo pair images are rectified, they are transformed so that their projection planes are coplanar. The coplanarity of the projection planes shifts the epipoles to infinity, making the epipolar lines horizontal. This simplifies the matching problem by reducing the search area along a single image row. With an additional transformation, the matching horizontal epipolar lines become collinear, meaning that the matching points in the stereo images have the same row coordinate. An example of a rectified stereo pair with the epipolar lines is shown in Fig. 2.4.

When a scene is captured from two different perspectives with the cameras whose parameters are known, Fig. 2.3, the epipolar constraint can be expressed by the *essential matrix*. The essential matrix is computed from the fundamental matrix using the following equation

$$E = K'^T F K \tag{2.13}$$

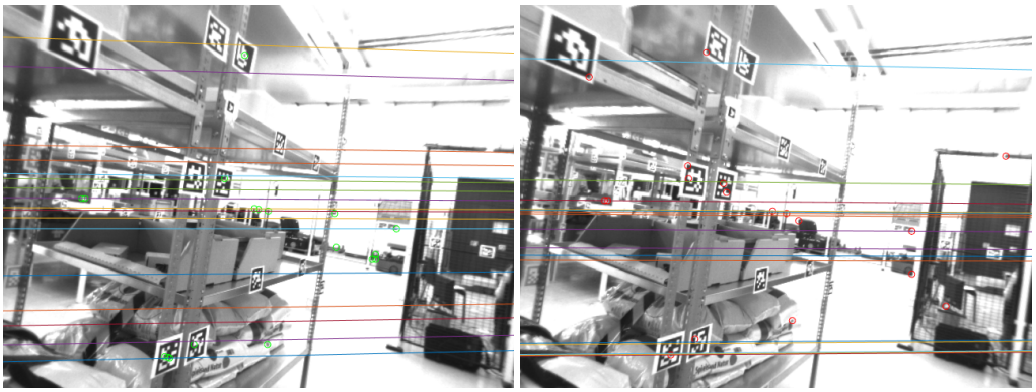
where K' and K are the calibration matrices of the cameras. The epipolar constraint from Eq. (2.12) is then expressed by

$$0 = x'(K')^{-T} E K^{-1} x \tag{2.14}$$

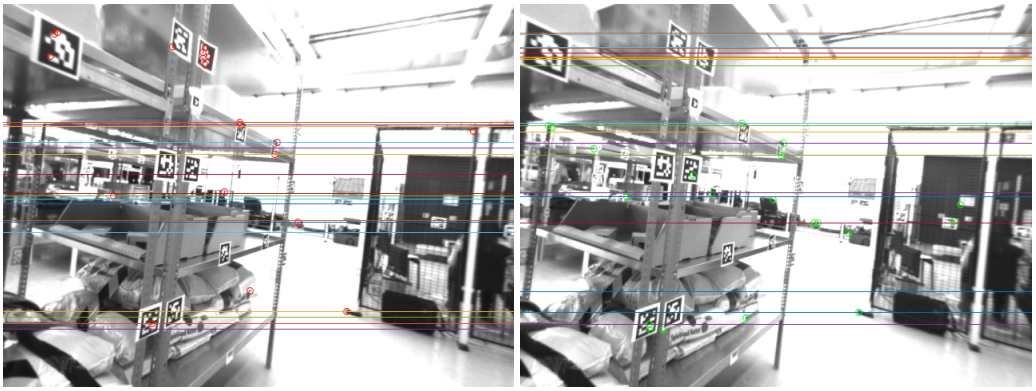
$$0 = \quad \tilde{x}' \qquad E \quad \tilde{x} \tag{2.15}$$

where $\tilde{x}'$ and $\tilde{x}$ are the points x' and x but on the normalized image plane in the camera coordinate frame. The essential matrix is useful in the multiview scene reconstruction and in visual odometry because it contains the information about the relative rotation and translation between the two views. However, estimating the relative transformation begins with matching points between the images. Since these points are expressed in the image coordinate frame, camera parameters are required to find the essential matrix.

DISPARITY. A stereo image pair projects the same scene into two images that are slightly different. The difference is due to the different perspectives and, the closer the scene is, the more obvious this difference in perspective becomes. As mentioned earlier, the position of a 3D point is determined by triangulation with known 3D point projections and



(a) Epipolar lines before rectification.



(b) Epipolar lines after rectification.

Figure 2.4: Comparison of epipolar lines before and after rectification. Before, the lines are slightly slanted (most obvious in top and bottom lines). After stereo rectification, the epipolar lines become horizontal. The points marked with red and green circles in one image have a match somewhere along the epipolar line in the other image.

the transformation matrix between the optical centers of the stereo pair. After rectification of the stereo image, the 3D point projections have the same row coordinate and different column coordinates.

The projection difference in the columns of the rectified images is called a *disparity* and is inversely proportional to the distance between a 3D point and the stereo pair. Equation (2.16), which relates the disparity and the distance between the 3D point from the cameras, comes from triangulating the matching projections of the rectified stereo pair.

$$d = f\frac{B}{z} \tag{2.16}$$

where f is the focal point of both cameras, B is the distance between the optical centers, called a *baseline*, and z is the distance from the common plane of the cameras. The visualization of the disparity values of all pixels in the image is called a *disparity map*.

The disparity computation is actually a stereo correspondence problem, i.e., finding the matching points in the left and right images. Assuming Lambertian surfaces in the scene, the correspondence problem is solved by finding two points in the images that are as similar as possible. However, finding the corresponding points is not always easy because of the following: (i) noise that changes the value of image pixels, (ii) textureless regions where

multiple points look similar, and (iii) occlusions where some points have no corresponding point in the other image.

The approaches with runtime limitations improve the disparity computation by focusing on a sparse group of points that can be matched unambiguously [16], [17]. In [16] authors use seed-growing method around the randomly sampled points in disparity space to compute the semi-dense disparity of rectified stereo pair, while authors in [17] used feature matching to estimate a sparse disparity map. In some cases, disparity computation is improved by estimating the disparity of larger area with sparse feature correspondences like in [18] where the SURF features are used to estimate the disparity of the segmented image areas.

Commonly, approaches are oriented towards *dense* or *area-based* correspondences, where the disparity is computed for almost all pixels in the image. These approaches are computationally more complex, but they provide more information about the scene. In the remainder of this section, we will focus on dense correspondence approaches.

The dense correspondence approaches can be divided into two categories

- Classic approaches, which have a structure of handcrafted blocks
- Modern, learning-based approaches, which use pre-trained neural networks

CLASSIC APPROACHES. The classic approaches are encompassed with a categorization given in the work of Scharstein and Szeliski [19]. According to them, the classic approaches have the following four steps:

- Matching cost computation
- Cost aggregation
- Disparity computation
- Disparity rafinement

Matching cost computation The first step of the disparity computation is common to all approaches. In this step, we determine the likelihood of two image patches being a match based on one of the similarity measures. The similarity measure takes patches centered at the pixel of interest in a reference and template images and computes the likelihood that they are a match in an intensity-based manner. The most popular similarity measures are *sum of absolute differences* (SAD), *sum of squared differences* (SSD), *normalized cross correlation* (NCC) [20], and *census transform* (CT) [21]. For patches $(2n+1)x(2m+1)$ centered at pixels (x, y) and $(x+d, y)$, the NCC, SAD, and SSD measures are given in Equations (2.17)-(2.19).

$$NCC(x,y,d) = \frac{1}{(2n+1)(2m+1)} \frac{\sum_{i=-n,j=-m}^{n,m} \left(I_l(x+i,y+j)-\mu_l\right)\left(I_r(x+d+i,y+j)-\mu_r\right)}{\sigma_l \sigma_r} \tag{2.17}$$

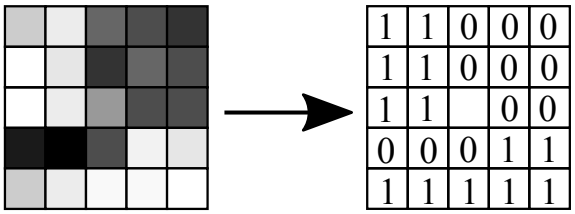


Figure 2.5: A visualization of census transform. An intensity of a center pixel is a reference value, darker pixels become zeroes and brighter become ones. The neighborhood is then reshaped in a binary word.

where μ_l,μ_r, σ_l and σ_r are respectively means and standard deviations of patches' intensity values in left and right images of a stereo pair.

$$SAD(x,y,d) = \sum_{i=-n,j=-m}^{n,m} |I_l(x+i,y+j) - I_r(x+d+i,y+j)| \tag{2.18}$$

$$SSD(x,y,d) = \sum_{i=-n,j=-m}^{n,m} (I_l(x+i,y+j) - I_r(x+d+i,y+j))^2 \tag{2.19}$$

The census transform makes a binary word of the patch in both images with the following formula

$$\begin{aligned} CT &= \bigoplus_{i=-n}^{n} \bigoplus_{j=-m}^{m} s\left(I(x,y), I(x+i,y+j)\right) \\ s(u,v) &= \begin{cases} 1, & u < v \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2.20}$$

where $\bigoplus_{i=-n}^{n}$ concatenates bits in a binary word. A similarity value is defined as a Hamming distance between two binary words. The CT mapping is visualized in Fig. 2.5.

The appropriate similarity measure is defined by use case requirements, whether it is runtime (SAD, SSD) or robustness to noise, calibration errors, etc. (CT).

Local approaches Local approaches compute the similarity measure for a window that moves in a predefined range of disparity values, assuming Lambertian surfaces in the scene, and choose the disparity value that best fits the local constraints. Such approaches, also known as *winner-take-all* (WTA) approaches, are fast but do not take into account the smoothness of the scene and the computed disparity map may have inconsistencies that reduce the quality of the final result.

Global approaches Global methods have a different approach to disparity computation. After the matching cost is computed, additional global constraints are added to account for possible errors of local constraints in difficult image areas (textureless or reflective surfaces). In the optimization step, the global methods compute the disparity map D by minimizing the global energy function in Equation (3.3). The first part of the energy function, E_{data},

is determined by local constraints, i.e., similarity measure, while the second part, E_{smooth}, propagates the smoothness constraint over the neighboring pixels.

$$E(D) = E_{\text{data}}(D) + E_{\text{smooth}}(D) \tag{2.21}$$

Global energy function minimization is performed using *graph cuts* (GP) [22], [23], *belief propagation* (BP) [24], or *dynamic programming* (DP) [25]. Even though the global methods preserve the consistency of the scene, their computation is NP-hard problem due to the global smoothing term.

Semi-global matching Semi-global matching (SGM), introduced by Hirschmüller [26], approximates global two-dimensional constraints by multiple one-dimensional constraints. More one-dimensional constraints improve the complexity of global approaches, but also maintain the consistency of the final disparity map. This method is popular for real-time applications [27], [28], [29] and we describe the steps of the SGM in more detail in Chapter 3.

LEARNING-BASED APPROACHES. In early attempts to incorporate learning techniques in solving the disparity problem, similarity measures were trained instead of using the hand-crafted measures [30],[31] while the SGM was used once the cost was calculated. Trained similarity measures showed better performance than hand-crafted ones [30], and they could also handle multiscale patches [32]. The improved performance of trained similarity measures is a result of optimization for a particular use case. In later experiments, networks were developed using an end-to-end approach [33], i.e., the networks did not rely on a hand-crafted solution in a later step, and were found to perform even better as they could be fully optimized for a particular use case.

Learning-based approaches made it possible to estimate depth, i.e., disparity, from a monocular camera. Although all previous depth estimation approaches were stereo based, depth can also be inferred using one image and the known sizes of the objects in the image [34]. Humans can also guess the depth of the scene with one eye closed since they learned the usual sizes of objects in space. Mono depth estimation is gaining popularity because it eliminates the need for prior extrinsic calibration of a stereo pair, which can get decalibrated after some time.

Nowadays, end-to-end approaches play a leading role on popular disparity benchmarks such as Middlebury [19], KITTI2012 [6], KITTI2015 [35], and SceneFlow [33]. On the other hand, training networks requires a lot of time, expensive hardware (GPUs), and training data. Networks trained using supervised learning also require a large amount of ground truth data, which can be expensive to generate. In addition, once trained, networks achieve their best runtime performance with GPUs. In-depth overviews of stereo and mono approaches can be found in [36] and [37], respectively.

VISUAL LOCALIZATION

The localization problem in robotics finds an answer to the question "Where is the robot?". This ill-posed problem has been repeatedly answered for decades and is still not fully solved [38]. In the last decade, visual localization approaches have experienced a breakthrough and

are now being actively explored in the robotics community [39]. The informative nature of visual sensors at a relatively low price makes them so popular.

Localization problem

LOCALIZATION. The robotics community has been actively working on the localization problem over the past few decades and has continuously improved the solutions so that they work in more complex scenarios. The complexity of the problem is defined by the amount of prior information needed for localization. Based on the prior information, localization can be divided into the following categories

- Localization based on artificial beacons in the environment
- Localization based on a prior map of the environment
- Localization without prior knowledge of the environment

The simplest scenario involves artificial beacons in the environment that are used to determine the pose of the robot. The active beacons exchange information about the distance between them, while the passive beacons, such as markers, are detected by a localization system. The best known example of localization using active beacons is the Global Navigation Satelite System (GNSS). In robotics, passive beacons in the form of reflective markers are often attached to robots to determine ground truth location with the OptiTrack motion capture system. Although approaches to localization using artificial beacons are relatively easy to implement, their use case is usually limited to small indoor areas (Optitrack) or outdoor-only areas (GNSS). Examples of fiducial beacon localization can be found in [40], [41], and [42], where the authors localized an agent in indoor environments using AprilTags, UWB sensors, and reflective markers, respectively.

A localization approach that requires a prior map of the environment is more complex, as the location is determined by overlaying the sensor measurements upon the map. This can be a difficult task since maps and sensors may be in different domains, e.g., if the map is created with a 3D LIDAR and a robot uses a stereo camera for localization. On the other hand, this localization approach provides more autonomy since it does not require manual installation of artificial beacons. There are different types of prior maps that can be metric [43], topological [44], or semantic [45]. Metric maps take care about the geometric relationships between elements in the map, and given the metric map one can compute distance between two elements of the map. The continuous metric maps represent the environment with geometric primitives like lines and planes [46]. The discretized metric maps are commonly represented with occupancy grids [47], which discretize the environment into cells which can be occupied, free or unknown. On the other hand, topological maps focus on the relations between elements of the environment, while the exact distance between the elements is irrelevant. The topological maps represent environment elements as nodes of a graph, and the relationship between the elements is represented with edges. The semantic maps augment the metric or topological map by providing the meaning to the map elements.

The most complex localization approach is the one that does not use any prior information. This approach is also known as *simultaneous localization and mapping* (SLAM), since the map is not known and the robot needs to map the environment while localizing itself in that map. This approach provides the highest level of autonomy but is still unsolved due to its complexity, although it has been in the focus of the robotics community for decades. The complex problem is divided into two parts: the front-end, which uses the sensor data to generate localization constraints, and the back-end, which uses the computed constraints to build a map and localize the robot.

In a simpler scenario, a map of the environment is known, and the robot needs to match measurements from its sensors with the known environment structure. Based on information fusion, localization approaches are classified into the following groups:

- filter-based approaches
- optimization-based approaches

Filter-based localization uses recursive Bayesian estimation to find the pose with the highest probability based on two functions, the *measurement model* and the *motion model*. The measurement model indicates which measurements to expect based on the assumed pose and map. In a simple 2D example: If we estimate the pose of a robot in an open space, we expect to measure no obstacles, while in a scenario where the robot is assumed to be near a corner, we expect to measure two perpendicular line obstacles. The motion model estimates the current pose of the robot based on its previous pose and the measurements taken between the previous and current poses. A popular example is wheeled odometry, where robots with wheels count the number of revolutions of their wheels and compute the distance traveled based on the known circumference of the wheel. The wheeled odometry is prone to error due to wheel slippage and variable wheel circumference. Filter-based localization transforms previous estimates of robot pose with the motion model and reweights the pose estimates with the measurement model. This process is constantly repeated and hopefully leads to the true robot pose after some time. Two popular examples of filters used for localization are the Extended Kalman Filter (EKF) [48] and the Particle Filter (PF) [49]. The EKF keeps track of the pose estimate in a form of Gaussian distribution, which is propagated through the motion and measurement models. The extended part of the filter deals with nonlinearities in the motion and measurements model which are linearized around the current state estimate. The EKF is computationally efficient when the size of the estimated states is small, but its limitation to unimodal distributions can in some cases degrade the localization performance. On the other hand, PF represents the pose estimate with a set of scattered samples. Each sample is one pose estimate, which changes its value with the motion model, and gets weighted with the measurement model. After the measurement model assigned weights to the samples, in the resampling step, particles with higher weights have more chances to reproduce, and thus the new samples are focused around the probable true pose. The PF is robust to nonlinearities and can represent multimodal distribution, but it is computationally more complex than EKF. A common property of all filtering approaches is that they work *online*, i.e., as soon as the measurements are used, they are incorporated in the filter estimate and there is no need to store older measurements for the next inference.

On the other hand, this means that we cannot remove the erroneous measurements, but only improve the estimate with additional correct measurements.

In the optimization approaches, all the constraints from the measurement and the motion models are saved and the location estimate is computed by minimizing the cost function when needed. This approach is *offline* because every time we want to compute the estimate, we can use all the data collected up to that point. The optimization approaches are computationally more demanding than filtering approaches if we want continuous pose estimation. The major advantage of optimization approaches over filtering approaches is their better robustness to nonlinearity. As mentioned earlier, the filter-based approaches incorporate all the measurements used, which are sometimes wrong due to linearization of motion and measurement models around the wrong point. This error cannot be removed from the estimate afterwards. In optimization-based localization, relinearization is possible because we keep all measurements.

Localization with visual sensors

VISUAL ODOMETRY. In the last decades, a camera sensor has become more popular in localization and SLAM problems. Starting from Moravec [50] who developed an obstacle avoidance system based on a slider stereo system, and Nistér [51], who developed an efficient 5-point algorithm for relative pose computation and coined the term visual odometry, numerous visual odometry approaches have been developed to improve robot's autonomy. Visual odometry estimates the motion of the robot based on a sequence of images captured by one or more onboard cameras. For two consecutive images in the sequence, visual odometry provides a spatial transformation of a camera that captured those two images. Although successive pose estimation accumulates estimation drift, the visual odometry approach accumulates it at a much lower rate than its predecessor, wheeled odometry. The approach is based on the assumption that the environment captured by a camera is static and the changes in the images are caused by the motion of the camera. Visual odometry is divided into:

- *direct* or *appearance*-based approaches, and
- *feature*-based approaches

Direct-based approaches look for a transformation between images by minimizing a photometric error between the images. The photometric error works directly with pixel intensities, hence the name direct approach. Based on the number of pixels used for minimization, there are dense methods [52], semi-dense methods [53], and sparse methods [54]. The direct approaches have better performance in scenes with poor texture, but are more sensitive to the unmodeled rolling shutter effect [54].

Feature-based approaches work with a set of sparse, locally distinctive areas of image called *keypoints* or *features*. By tracking the keypoints in two images captured by a moving camera, a pose transformation between these two images can be computed by minimizing the reprojection error. The keypoints are tracked through the images using similarity metrics of their *descriptors*. The descriptors are vectors computed using the intensity values of pixels in a patch centered at the keypoint's coordinates. For robust tracking, it is desirable that the

descriptors of the same keypoint match regardless of image rotation and different image scales. Several feature detectors have been developed for different use cases. SIFT [55], for example, is very robust and invariant to rotation and scaling, but slower than other detectors. On the other hand, FAST [56] is computationally efficient but not as robust.

In this thesis we used feature-based method SOFT [57] [58], which is described in more detail in Section 4.3.

VISUAL LOOP CLOSURE DETECTION. Visual odometry provides a local pose estimate that is degraded by successive estimates. For long-term localization, the accumulated odometry drift needs to be corrected. In the SLAM problem, the corrections come from revisiting a previously mapped area and detecting that location, called *loop closure detection*. After a loop closure detection, the SLAM problem corrects both the pose and the map estimates, while the localization corrects only the pose, since the map is not the element of the estimation process.

In man-made environments, localization and SLAM can be improved by placing distinctive fiducial markers. The authors in [59] developed an approach for visual SLAM that uses AprilTag markers [60] to improve pose estimation, called TagSLAM. The TagSLAM can improve both the map and the trajectory because both tracking and loop closure are based on unique, easily detectable fiducial markers. Furthermore, the TagSLAM can take the built map of the environment and provide estimated pose of the camera based on tracking and loop closing. However, for larger areas that require a large number of tags, or for outdoor areas with many uncontrolled factors, a different loop closure approach is required. The classic approach for visual loop closing is matching of features in the current image with the features stored in the map. Checking all features in the map would be time consuming and not feasible for online applications. Thus, the image information is transformed into a descriptor with a higher level of abstraction, which speeds up the comparison of the current frame with the loop closure candidates. Authors in [61] presented DBoW2 which transforms image BRIEF descriptors [62] into words using a hierarchical vocabulary tree. However, the vocabulary tree is built before the online application using the training images. Similarly, authors in [63] presented a low memory cost approach based on a simplification of the Fischer vector. In contrast, learning-based approaches compute the similarity directly on the images using a pre-trained convolutional neural network (CNN). In [64] authors showed that the CNN approaches have similar performance to approaches with hand-crafted descriptors, but are more robust when a change of illumination appears.

VISUAL SLAM. Visual SLAM is able to maintain long term localization consistency because in addition to visual odometry, visual SLAM also creates a map of the environment. This map provides accumulated odometry error correction once a loop closure happens. In addition, in case of a feature tracking failure, the visual SLAM is able to relocalize and continue with localization and mapping. There are several popular visual SLAM approaches among which the popularly referenced are MonoSLAM [65], DSO [66], and ORB-SLAM2 [67]. MonoSLAM is one of the first mono camera SLAM solution that estimates the trajectory and a map of features in real-time using the EKF. The sparse direct approach which estimates the trajectory and builds the map is DSO. Although stereo DSO does not perform loop

closures, its results on public datasets are outperforming some of the state-of-the-art SLAM methods. Later, an extended version for mono camera LDSO [68] also included loop closing. The ORB-SLAM2 is one of the current state-of-the-art visual SLAM solutions with open-source code and since it is used in the evaluations in the following chapters, we provide an overview of the method in the sequel.

The ORB-SLAM2 method consists of three main parts: Tracking, Local Mapping, and Loop Closing, which are executed simultaneously in separate threads. The method is designed to work with mono, stereo, and RGB-D cameras, and while variations of the method exist depending on the camera used, the focus here is on the stereo camera. The keyframe-based approach of ORB-SLAM2 splits the map into smaller segments that are connected in a *covisibility graph*. Local *bundle adjustment* (BA) corrects both camera and feature poses in the current keyframe and its neighboring keyframes in the covisibility graph, making the optimization process independent of map size. However, the ORB-SLAM2 uses keyframes to simultaneously build an *essential graph* that is used for global pose optimization when loop closure is detected. The essential graph has fewer edges than the covisibility graph, which improves the runtime of the pose graph optimization. After the pose graph optimization, global consistency is achieved by an additional full BA in a separate thread.

Tracking The ORB-SLAM2 is a feature-based method and uses ORB features [69]. After the first step, where only the first key frame is created, all other steps are matched with the features from the previous frame and the initial estimate of the camera pose is computed using the motion-only BA. With the initial pose estimate, features from the local map (the neighborhood in the covisibility graph) are projected onto the current frame, and the pose is optimized again using the features from the local map. In the final step of tracking, it is decided whether the current frame is informative enough to serve as the new keyframe.

Local Mapping After the decision to insert a new keyframe is made, the covisibility and essential graphs are updated. This means that a new node is added to the graph and the edges of the graph are updated. Then, ORB-SLAM2 tests the robustness of the points recently added to the graph and removes the problematic ones. It is assumed that the remaining points are robust. Using the features of the new keyframe, new points are added to the map and all points and neighboring keyframes in the covisibility graph are optimized through the local BA. Finally, all redundant keyframes, i.e., those whose points are already seen in other keyframes, are removed to improve the complexity of bundle adjustment.

Loop Closing The loop closing detects the same map elements between the nonconsecutive keyframes and reduces the accumulated pose drift and map deformation. The ORB-SLAM2 checks for potential loop closures using bag-of-words place recognition based on DBoW2 [61]. Loop closure candidates are then further filtered through a matching process. If sufficient matches are found, a loop closure is accepted and covisibility and essential graphs are updated accordingly. In case of tracking failure, ORB-SLAM2 is able to relocalize in the existing map using the DBoW2 place recognition module.

A detailed description of the method can be found in [70] and [67]. Although there is a newer version ORB-SLAM3 [71], the proposed solutions are compared with ORB-SLAM2,

since the newer version provides special features for map building in case of tracking failure. In case of tracking failure, the ORB-SLAM3 starts creating a new map and a new trajectory, which is eventually merged with the older map once the loop closure is detected. However, in initial tests on our dataset, the ORB-SLAM2 managed to compute a more compact trajectory and is therefore used in the evaluation.

DATASETS FOR EVALUATION

The implemented methods were evaluated on five publicly available datasets. The KITTI dataset [6] is widely used in the robotics community for various estimation problems and we found it suitable for evaluating the method described in Chapter 3. In addition, the dataset KITTI MOD is used for evaluating the moving object detection method described in Chapter 3.

The Dortmund, Augsburg, and Zagreb datasets were recorded specifically for the evaluation of our methods described in Chapters 4 and 5 because they required specific localization information that was not provided in the available online datasets.

KITTI and KITTI MOD datasets

The KITTI dataset was recorded in Karlsruhe, Germany, and contains images of urban, and rural, areas as well as highways, taken with a set of sensors mounted on a mobile platform. The dataset contains sequences recorded with two color cameras, two monochromatic cameras, and a Velodyne laser scanner. The dataset contains benchmarks for various estimation problems such as odometry, optical flow, disparity, segmentation, and more. In addition, the dataset also includes images of all sequences in two versions: unrectified-unsynchronized and rectified-synchronized. Ground truth data is computed with measurements from onboard GPS and the Velodyne laser scanner.

The *Sceneflow* and *Stereo* benchmarks were unsuitable candidates for evaluating our disparity computation method because they both contain a set of stereo image pairs from different sequences with no temporal correspondence, whereas the main advantage of our method is to pass disparity information across the image sequence. Therefore, we used parts of rectified synchronized sequences[1]. The scenes were carefully selected to maximize the continuous sequence parts without moving objects in the scene. Only the 2011_09_26_drive_0051_sync sequence contains moving objects and serves to show the limitations of our method. The sparse ground truth was captured with the Velodyne laser scanner.

The KITTI dataset does not provide data for training and evaluation of moving object detection methods. However, they do provide the 3D object labels for the *3D object detection* benchmark. In [72] authors used these labels to extend the KITTI dataset with moving object detections and named it KITTI MOD dataset. The existing detections of objects are projected onto the 2D bounding boxes and the corresponding bounding boxes are

[1] Full names of the used sequences are: 2011_09_26_drive_0035_sync, 2011_09_26_drive_0039_sync, 2011_09_26_drive_0051_sync, 2011_09_26_drive_0061_sync, 2011_09_26_drive_0086_sync, 2011_09_26_drive_0093_sync, 2011_09_26_drive_0117_sync

associated over the sequence. The movement of the bounding boxes is computed based on the pose change between two associated bounding boxes. The ground truth camera motion is then subtracted from the detected bounding box motion to determine if the object in the bounding box is moving independently. The dataset KITTI MOD contains labeled bounding boxes around the vehicles that are classified as static or moving. The detections are available for the images from the KITTI *Sceneflow* benchmark.

Self recorded datasets

Most of the public datasets focus on one of the classic problems in robotics such as SLAM, visual odometry, disparity, optical flow, semantic segmentation, etc. When these problems are applied to a particular use case, there are often additional constraints that can be used to improve the solution. On the other hand, the solutions that exploit the use case constraints may not be regularly evaluated on available datasets. The method described in Chapter 4 and Chapter 5 is an example of such a solution that exploits the use case constraints. The method performs localization based on a fusion of information from a stereo camera and a monocular camera. The localization depends on the known poses of ground markers distributed in the warehouse-like environment. These very specific requirements made all found public datasets inapplicable for the evaluation.

For this reason, we have recorded three datasets: *Dortmund dataset* [2], *Augsburg dataset*[3], and *Zagreb dataset*[4]. In all datasets, a camera sensor suite, Fig. 2.6, is attached to the back of the Safety Vest. The camera suite consists of a horizontally-oriented stereo camera, PerceptIn Ironsides, and a downward-looking monocular camera, Chameleon3 CM3-U3-50S5M- CS with a Computar, 12 mm, 2/3", 5 MP lens. The cameras are installed on a thick aluminum metal plate that ensures rigid and fixed displacement between them. Each dataset is taken in a different warehouse-like environment and has a known map of ground markers distributed through the localization area. An example of ground markers used in the datasets is shown in Fig. 2.7. The appearance and size of the ground marker correspond to the markers used by the CarryPick robots in Swisslog's warehouses[5].

The method is based on the fusion of information from the stereo camera and the monocular camera. The fusion requires known transformation matrices between the cameras in the stereo pair and the monocular camera. In the Dortmund dataset, the stereo camera and the monocular camera shared a very small overlapping part of the scene, which made the extrinsic calibration prone to errors. Extrinsic estimation is improved by adding an additional camera, the Intel RealSense D435, which has sufficient overlap with both cameras, Fig. 2.8. Although the RealSense D453 has 3 cameras, we used only one camera for calibration. In this way, instead of having one ambiguous transformation matrix, we had two more reliable transformations that together gave the transformation we were looking for. The extrinsic parameters of the cameras used in datasets are computed with Kalibr calibration package [73].

[2] https://zenodo.org/record/4456051
[3] https://zenodo.org/record/4456723
[4] https://zenodo.org/record/4456062
[5] The Swisslog company is one of the partners on the project SafeLog.



Figure 2.6: Safety Vest with the sensor setup that consists of an IMU-aided stereo camera and a downward-looking monocular camera. This placement of the sensors was chosen since it will not disturb the human when performing the usual tasks. Furthermore, cameras cannot get obstructed by hands, and this part of the human body is the most stable and has the smallest chance of doing abrupt motion that could blur the images.

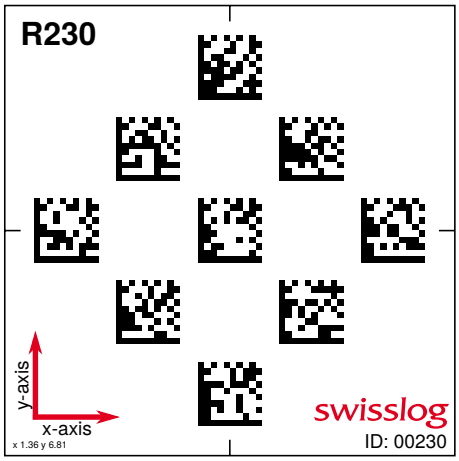


Figure 2.7: The fiducial marker used for localization of CarryPick robots in the Swisslog's warehouses. Each marker has a special ID and a different combination of the 9 rectangular patterns, so-called DataMatrix.

DORTMUND DATASET. The Dortmund testing area was approximately 5×3 m^2 and equipped with the Optitrack motion capture system. The recording area contained one real rack, while other racks and walls were imitated with plastic boxes, as shown in Fig. 2.9. The height of the boxes was set to allow the motion capture sensor to record the ground truth pose of the camera suite and prevent the scene out of the arena to be present in the camera's field of view. There were 6 ground markers on the floor of the testing arena and their pose was measured using the Optitrack system. The dataset contains 9 different sequences of a human performing typical tasks such as walking and bending between the racks. In each sequence, a human initiates the localization process by standing over the ground marker at the entrance of the arena. Accordingly, localization ends when the human returns to the starting ground marker. During the sequence, the environment is static, which only partly



Figure 2.8: RealSense D435 camera placed in between viewing areas of stereo camera and monocular camera to improve extrinsic calibration. Only a color camera was used during the calibration.



Figure 2.9: The experimental arena for collecting Dortmund dataset. The plastic boxes imitate racks and walls. The arena is covered with Optitrack motion capture system. (By courtesy of Fraunhofer IML).

simulates the scenario of a real warehouse, where the racks are moved by the robots that carry them to the picking stations and back. Simulating a scenario with moving racks was not feasible during the recording of a sequence. Therefore, the dynamic environment was simulated by merging the first five sequences that contained different floorplans. Since the start and end positions were the same in all sequences, a merged sequence contains very small pose jumps in transitions between sequences.

AUGSBURG DATASET. The Augsburg data set is a close approximation of a real warehouse scenario. The size of the warehouse testing arena was approximately 12×13 m^2 and was filled with metal racks as shown in Fig. 2.10. The biggest challenge in dataset acquisition was



Figure 2.10: The testing arena of Augsburg dataset. This arena is normally used to test algorithms on robots before the deployment, and its appearance (racks, ground markers, safety fence) is closely similar to the real warehouse.

capturing the ground truth of the human's pose. Unlike the Dortmund arena, the testing facility arena was not equipped with a motion capture system. After considering all the limitations, we decided to use two approaches to evaluate localization. The first approach was to mark several control points on the floor of the testing arena, whose position was measured manually, and walk over these points during the experiment. The second approach was to accurately compute the location of the cameras using AprilTag markers [60], which we additionally installed on the racks, Fig. 2.11. We used TagSLAM [59] to obtain the map of the markers.

To ensure the most accurate AprilTag map (with the position and orientation of the markers), we manually measured the position of all markers used to initialize the SLAM algorithm. The mapping was performed in a separate experiment where we focused solely on the detection of the AprilTag markers. Once the map was created, we used it to compute the localization ground truth data for subsequent experiments. It proved infeasible to cover the entire test site densely enough with AprilTag markers to obtain smooth ground truth data for the entire trajectory of the Safety Vest, as the optimization became unstable and crashed during the map generation process. Therefore, we focused on two sections of the arena for which we could obtain a reliable ground truth estimate. We assume that the accuracy of ground truth pose in these sections is less than 20 centimeters. This dataset is divided in three scenarios. The first is the standard operating conditions scenario, which contains 4 sequences with the human walking in the warehouse and performing typical tasks. The second scenario is the kidnapped-human scenario, in which cameras were briefly covered to simulate a situation where the sensors' field of view is obstructed and localization fails. Finally, the third scenario is the case of a non-static environment where the racks have been redistributed during the sequences.

ZAGREB DATASET. The Zagreb dataset is used to evaluate the improved localization method in warehouse environments that uses a network of UWB sensor network in addition

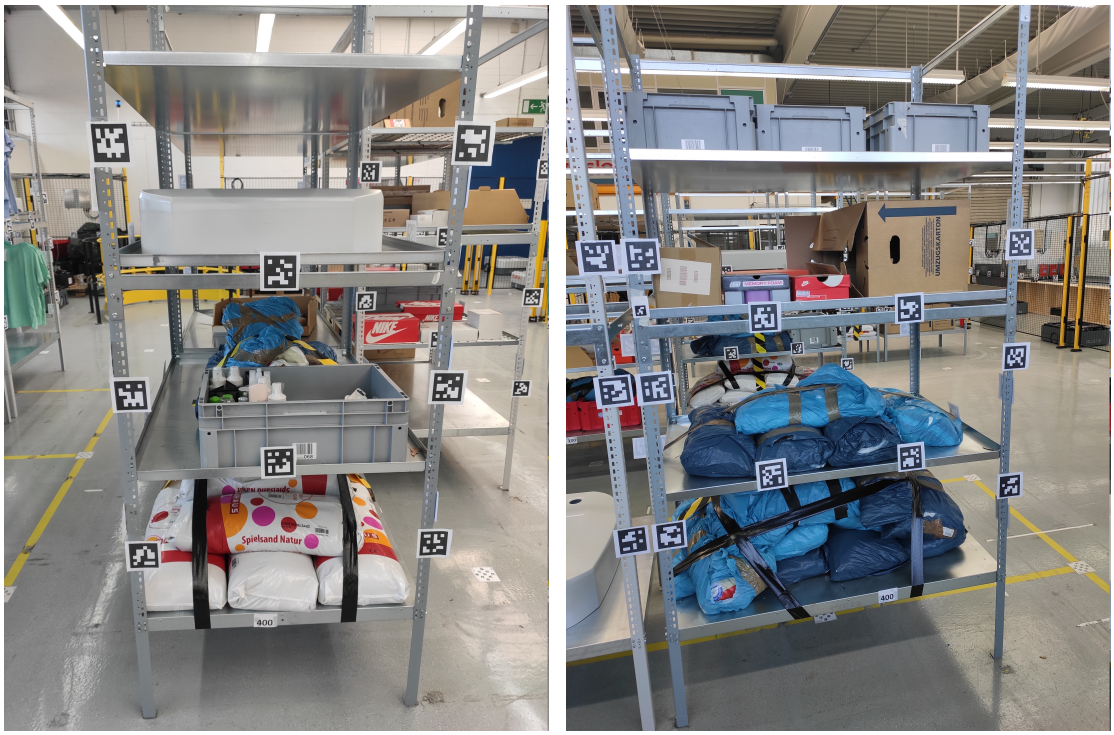


Figure 2.11: AprilTags placed throughout the testing arena for acquiring the ground truth pose information. All AprilTags are unique and their map is computed with TagSLAM package.

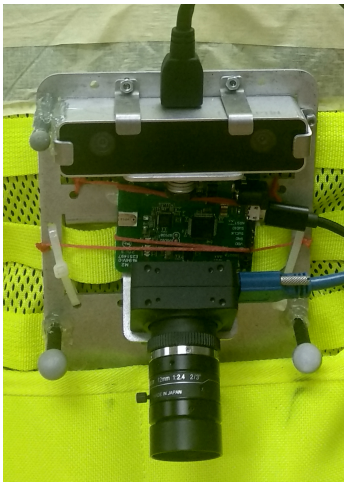


Figure 2.12: The visual sensor suite, stereo and monocular camera, upgraded with the UWB sensor.

to visual sensors to improve localization in visually challenging situations. The sensor suite is extended with a UWB node[6], Fig. 2.12, that performs ranging measurements to other nodes in the UWB network. For the extrinsic parameters of the UWB node, we used an approximation in which we assumed that the position of the UWB node coincides with the position of the left camera in the stereo pair. The placement of the UWB node was close to the stereo camera, as shown in Fig. 2.12, and the UWB ranging accuracy and the accuracy required for our use case justify this approximation.

The dataset was recorded in a faculty library that visually resembles a warehouse envi-

[6] Pozyx's Creator kit (https://www.pozyx.io/)

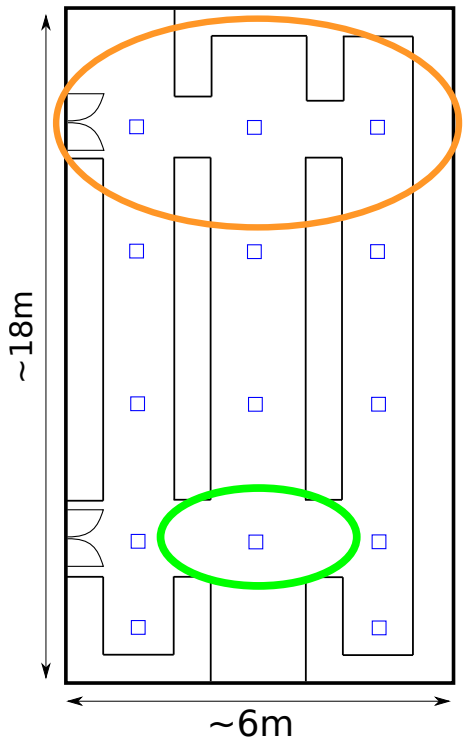


Figure 2.13: The floorplan of the faculty library in the Zagreb dataset. Blue squares mark the position of ground markers, the orange ellipsoid marks the area with presence of UWB anchors, and the green ellipsoid shows the area where the ground truth was available.

ronment due to the ceiling-high metal bookshelves and long, narrow corridors. The library floorplan is shown in Fig. 2.13. From this we can see that the area is approximately 6×18 m^2 and consists of three long corridors with two passages at the end of each corridor. In the testing area, there are 14 ground markers, depicted as blue squares, evenly distributed throughout the library. The number of UWB sensors (five anchors) limited the area of UWB location cues to the part of the library marked with the orange ellipsoid. The anchors were placed to maximize coverage area, provide anchors coordinate diversity, and line-of-sight between the node on the Safety Vest and the anchors. Ground truth was acquired with the Optitrack motion capture system, which limited the availability of ground truth to the area within the green ellipsoid due to the narrow corridors and high shelves in the library.

The dataset contains 11 recordings. Each recording contains a different dominant type of motion. The descriptions of the dominant motions for each recording can be found in Table 2.1. The sequences have different levels of localization difficulty ranging from the easy sequences with slow walk through the corridors to the difficult sequences with fast lateral movements. Furthermore, sequence FL05 simulates a scenario where the visual input is disabled during the localization process. The layout of the bookshelves in the sequences could not be changed so the non-static environment is not evaluated on this dataset.

Evaluation metrics

In the following chapters, two different metrics are used to evaluate the proposed methods: (i) metrics for disparity evaluation and (ii) metrics for trajectory evaluation. The KITTI

Table 2.1: The list of dominant movements per recording in the Zagreb dataset.

Recording	Movement type
FL01	recording of all markers
FL02	normal speed walk through the library
FL03	slow walk with occasional crunching
FL04	slow lateral walk
FL05	normal speed walk with loss of images in stereo camera
FL06	normal walk, dominantly around the UWB area
FL07	long normal speed walk
FL08	fast walk
FL09	fast walk and rotating during the walk
FL10	long fast walk
FL11	running through the library

Stereo 2015 benchmark compares the methods based on the number of pixels with false disparity value, i.e., the percentage of outliers. The value of a pixel is false if the absolute difference between the estimated disparity and the actual disparity is more than 3 pixels and if the relative difference is more than 5%. The absence of the disparity estimate in places where the ground truth is available, also counts as a falsely estimated disparity value.

The trajectory is evaluated using the *absolute trajectory error* (ATE) metrics [74] provided in the package *rpg_trajectory_evaluation* [75]. Before computing ATE, the segments of the estimated trajectory are associated with the ground truth using the timestamps and are aligned with the ground truth to obtain the best fit. Then the ATE is computed using the following equation

$$\begin{aligned} \mathrm{ATE} &= \left(\frac{1}{N} \sum_{i=0}^{N-1} ||\Delta p_i||^2 \right)^{\frac{1}{2}} \\ \Delta p_i &= p_i^{\mathrm{GT}} - R_i^{\mathrm{GT}} \left(R_i^{\mathrm{EST}} \right)^T p_i^{\mathrm{EST}} \end{aligned} \tag{2.22}$$

The well-known shortcoming of the ATE is its sensitivity to the timing of error occurrence. *Relative pose error* (RPE) is commonly used to provide more informative statistics about the errors in parts of the trajectory. However, the RPE is not used for evaluation since in the most of the recorded sequences the ground truth is scarcely present.

3

Disparity estimation using stereo visual odometry

The autonomy of mobile systems is highly dependent on the amount of timely information about the environment. Currently, visual sensors are the most popular choice for autonomous systems. They have proven to be highly informative, and as computing power increases, information processing meets the requirements for real-time applications. One representation of visual information is the disparity presented in Chapter 2, which is a two-dimensional representation of the three-dimensional shape of the world. Dense disparity estimation in real-time is of great importance for the navigation of autonomous mobile robots.

Typically, disparity maps are computed on image sequences, and reusing previously computed disparity maps could improve the computation of future disparity maps in terms of accuracy, density, and computation time. The continuous fusion of disparity information through the sequence requires a recursive estimation filter. In [76], the authors introduced a pixel-wise Kalman filter and used it to estimate the dense disparity map of the scene captured with a monocular camera. Later, in [77], the authors also used a pixel-wise Kalman filter to improve the disparity maps on the sequences and tested the Kalman filter framework with different similarity cost functions and disparity computation approaches.

We present the computationally efficient SGM method based on the reduced disparity search space, first introduced in [78]. The disparity search space is reduced by passing the disparity information from the previous step and searching for a new disparity only in the uncertainty region around the expected disparity value. Both the expected disparity value and the uncertainty come from a prediction step of the pixel-wise Kalman filter. The uncertainty is adjusted for the evaluation dataset by estimating the process and measurement noise of the Kalman filter using the training sequences of the dataset. In addition, the expected disparity value is based on the assumption that all motion is from an onboard stereo camera. This made it possible to detect moving objects in the areas where the measured disparity is far from the expected value. A similar idea for moving object detection is presented in [79], where the authors used a similarity measure between temporally wrapped previous disparity and the current disparity to detect moving objects in areas with low similarity. Moreover, the runtime of SGM with reduced disparity search space is improved by implementing the method with single instruction multiple data (SIMD) instruction set and multi-threading.

The application scenario of the method is motivated by the SafeLog project, where the idea is that a human worker wears a stereo camera, a processing unit, and a power supply

during his usual work operations in the warehouse. During these operations, other workers or robots may come into close proximity, and detecting them might be useful information. The specific shape of the warehouse environment without a suitable training dataset and the limited computing and power resources led to the SGM solution with reduced search space. Also, due to the lack of ground truth in self-recorded datasets, the evaluation is limited to datasets with stereo images and ground truth available for image sequences, such as KITTI dataset.

In the remainder of this chapter, we first give an overview of the classic SGM method presented by Hirschmüller in [80]. Second, we introduce the idea of the reduced disparity search space and its implementation in the SGM with the Kalman filter. Since the Kalman filter requires values for process and measurement noise, which may change when using different methods in the prediction and measurement steps, we present our procedure for their estimation using the KITTI dataset. We then introduce the idea of detecting independently moving objects on the image sequence captured with an onboard stereo camera. In the last part, we show the results of the evaluation on the KITTI and KITTI MOD datasets and comment on the performance of the proposed methods.

CLASSIC SGM

The classic SGM, introduced by Hirschmüller in [26], has shaped approaches to computing disparities and is often used by the image processing community as the basis for their implementations [30], [81], [82], [83]. The method is subject to the classification of Richard and Szelinski and is divided into four steps: (i) matching cost computation, (ii) aggregation of costs, (iii) disparity computation and (iv) refinement of disparity.

Matching cost computation

The original similarity measure in [26] is mutual information. This measure, introduced in [84], is based on information theory and seeks a disparity map that maximizes mutual information. The input to the matching process is two images: a reference image and a template image wrapped with the disparity map. For notational simplicity, $I_r(\mathbf{p})$ is an intensity value of the reference image at pixel $\mathbf{p}$ and $I_t\left(D(\mathbf{p})\right)$ is an intensity value of the template image at pixel $D(\mathbf{p})$ wrapped with the disparity map D. Since the input images are rectified, warping is given by the following equation

$$D(\mathbf{p}) = \mathbf{p} + (d, 0) \tag{3.1}$$

where d is a disparity value at pixel $\mathbf{p}$. The mutual information MI_{I_r,I_t} is expressed as

$$MI_{I_r,I_t} = H_{I_r} + H_{I_t} - H_{I_r,I_t} \tag{3.2}$$

where H_{I_r} and H_{I_r,I_t} are entropy and joint entropy values, respectively.

When the disparity map, i.e., the warping function, is correctly estimated, the reference image has much information in common with the wrapped template image and their joint entropy in Eq. (3.2) has a small value. In other words, the correct disparity map maximizes the mutual information. One problem with MI is that the disparity map is

required in advance. This problem is solved by determining the disparity map iteratively and hierarchically. In this process, the resolution of the disparity map is increased in each step, and each step is initialized with the values of the disparity map from the previous step.

Cost aggregation

Global methods compute the disparity map D by minimizing Eq. (3.3), introduced in Section 2.

$$E(D) = E_d(D) + E_s(D) \tag{3.3}$$

The matching costs from the previous step are used to compute E_d, while E_s provides additional constraints to provide global consistency of the disparity map. The minimization of Eq. (3.3) written in a sum over pixel fashion becomes

$$D = \arg\min_D \sum_{\mathbf{p}} \left(C(\mathbf{p}, D(\mathbf{p})) + \sum_{\mathbf{q} \in \mathbf{N_p}} \begin{cases} 0, & D(\mathbf{p}) - D(\mathbf{q}) = 0 \\ P_1, & |D(\mathbf{p}) - D(\mathbf{q})| = 1 \\ P_2, & |D(\mathbf{p}) - D(\mathbf{q})| > 1 \end{cases} \right) \tag{3.4}$$

where $\mathbf{q}$ is a pixel in the neighborhood $\mathbf{N_p}$ of pixel $\mathbf{p}$. Penalties P_1 and P_2 are paid for having a different disparity value from the pixels in the neighborhood $\mathbf{N_p}$. The penalty P_1 allows disparity adaptation to surfaces whose depth is gradually changing, while the stronger changes will be controlled with P_2.

The minimization of such a problem is an NP-complete problem. Thus, the disparity is found with dynamic programming along individual one-dimensional paths in the image which make the problem solvable in polynomial time. The constraints from paths in different directions are combined by aggregating costs from all directions. For a direction $\mathbf{r}$, a loss function $L_\mathbf{r}(\mathbf{p}, d)$ accumulates cost for a pixel $\mathbf{p}$ with a disparity d by using the following formula

$$L_\mathbf{r}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min \begin{cases} L_\mathbf{r}(\mathbf{p} - \mathbf{r}, d) \\ L_\mathbf{r}(\mathbf{p} - \mathbf{r}, d \pm 1) + P_1 \\ \min_i L_\mathbf{r}(\mathbf{p} - \mathbf{r}, d + i) + P_2 \end{cases} \tag{3.5}$$

The first term in Eq. (3.5) adds the matching cost $C(\mathbf{p}, d)$ for the current pixel, while the previous loss is propagated with the second term. The second term has three possible options: (i) keep the same disparity value and take the accumulated loss, (ii) change disparity by a small amount and add penalty P_1 to accumulated loss, and (iii) make a jump in disparity and add penalty P_2 to accumulated loss. The aggregated cost $S(\mathbf{p}, d)$ is then computed by summing the loss $L_\mathbf{r}(\mathbf{p}, d)$ from all directions Eq. (3.6) (originally 16 directions are used).

$$S(\mathbf{p}, d) = \sum_\mathbf{r} L_\mathbf{r}(\mathbf{P}, d) \tag{3.6}$$

Edge-preserving is important for the quality of the disparity map. Since object edges often coincide with a disparity discontinuity, the penalties are adapted with the information

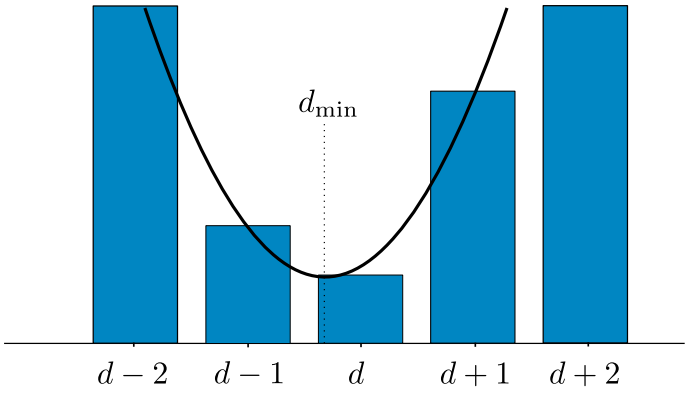


Figure 3.1: Fitting a quadratic curve on neighboring costs of a disparity d at a pixel $\mathbf{p}$. The minimum of a curve gives a subpixel disparity estimate $d_{\min}$ of pixel $\mathbf{p}$.

from intensity images. The adaptive P_2 value takes care of the edges by having a lower value in the areas with a high-intensity gradient, Eq. (3.7).

$$P_2 = \min\left(\frac{P_2'}{|I(\mathbf{p}) - I(\mathbf{p}-\mathbf{r})|}, P_1\right) \tag{3.7}$$

The P_2 value will always be higher or equal to P_1. The upper bound is not set because if $I(\mathbf{p})$ and $I(\mathbf{p}-\mathbf{r})$ are too close, one of two other options will be selected in second member of Eq. (3.5).

Disparity computation and refinement

The disparity map is selected with a simple winner-takes-all method by using a disparity value with minimal aggregated cost, Eq. (3.8).

$$D(\mathbf{p}) = \arg\min_d S(\mathbf{p}, d) \tag{3.8}$$

Values of disparity map D are integers since the change in cost from the previous steps is possible only for discrete per-pixel moves. The subpixel estimation can be achieved by fitting a quadratic curve through costs neighboring disparities, Fig. 3.1.

The remaining inconsistencies in disparity can be filtered with a small-sized (3×3 pixels) median filter. Additionally, a left-right consistency check can be performed by switching the reference and template images and repeating the disparity estimation process. The previously computed disparity D_a and newly computed disparity D_b are then compared with Eq. (3.9), and set to invalid value if the difference is higher than tolerance t.

$$D_a(\mathbf{p}) = \begin{cases} D_a(\mathbf{p}), & |D_a(\mathbf{p}) - D_b(D_a(\mathbf{p}))| \le t \\ \text{invalid}, & \text{otherwise} \end{cases} \tag{3.9}$$

REDUCED DISPARITY-SEARCH SPACE SGM

Semi-global matching showed improved runtime compared to global approaches. For use cases with high runtime demands and limited computational power, such as onboard computers, an efficient method for estimating the disparity map would be of utmost importance.

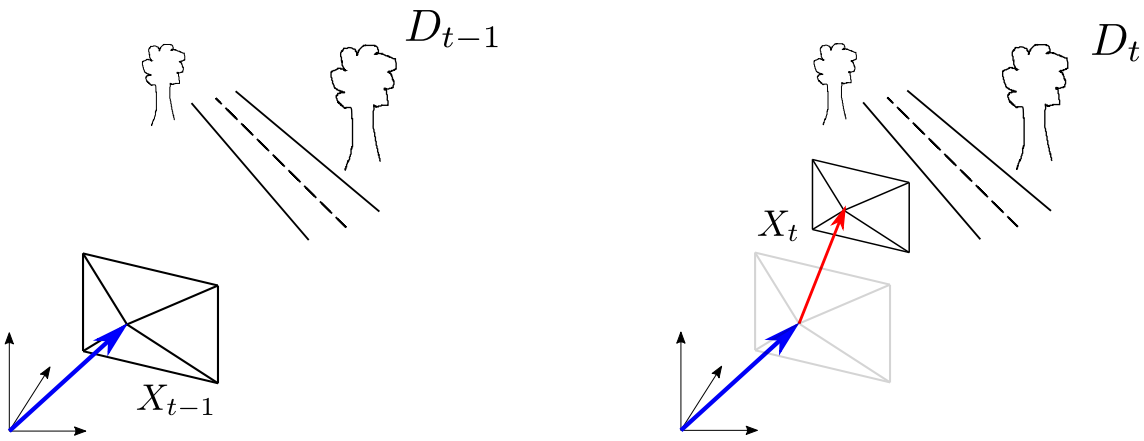


Figure 3.2: A scene recorded in two subsequent moments with a moving camera. Under the assumption that the environment is static, known pose change from X_{t-1} to X_t can be used to transform disparity D_{t-1} to D_t.

Possible modifications, such as hierarchical disparity computation and usage of SIMD instruction set, that can improve the complexity and memory consumption are proposed in [80].

Considering the possible improvements, our method and implementation focus on computational efficiency in all steps of SGM. A sequence captured by a moving stereo camera has a lot of common information about the scene between pairs of images from different time steps, i.e., the geometry of the scene. The main idea behind the approach is to use ego-motion information estimated with stereo visual odometry to wrap the disparity map from the previous step into the current step, Eq. 3.2. This wrapped disparity map is the expected disparity map for the current time step.

Passing information between steps is implemented with a pixel-wise Kalman filter. The SGM framework has been modified to allow the inclusion of prior disparity information in the estimation process. Except for the stereo pair images, the SGM gets upper and lower bounds of the disparity values for each pixel. In the classic method, these bounds are set to the predefined values for all pixels. The variable bounds reduce the disparity search space by removing the improbable disparity values. In the following subsections, we give an introduction to the Kalman filter and then explain the steps of the proposed method, the framework of which is visualized in Fig. 3.3.

Kalman filter

The Kalman filter is a linear quadratic estimator with a wide range of applications in autonomous and semi-autonomous systems. The filter recursively estimates the states of the system based on a sequence of user inputs and error-prone measurements. A system has one or more internal states that cannot be measured directly but are reflected in the system's behavior. Moreover, the problem becomes even more complex because these internal states can change over time. The changes in the system states are given by the *process model*. If the initial states $\mathbf{x}_0$ are given and we know the process model $\mathbf{F}$ of the system, the input model $\mathbf{B}$ that defines the effects of user input on the states of the system, and the sequence of inputs

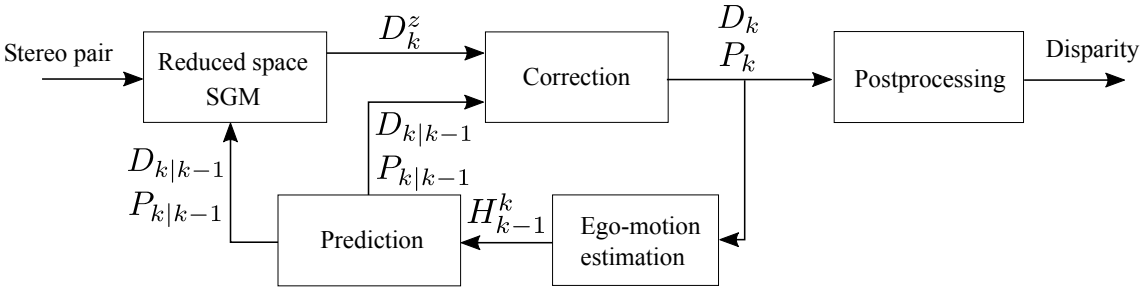


Figure 3.3: Block diagram of the proposed SGM algorithm. At the beginning, the classic SGM is computed in the full search space. Otherwise, the ego-motion estimation predicts the disparity map from the values of the previous step. The reduced disparity search space SGM block uses a stereo image pair and the disparity and variance prediction to compute the new disparity D_k^z. The computed disparity map D_k^z and the predictions $D_{k|k-1}$ and $P_{k|k-1}$ are used to correct the disparity map and variance. Outside the loop, the disparity is post-processed to improve the appearance of the final disparity map.

$\{\mathbf{u}_1, \cdots, \mathbf{u}_k\}$, the states of the system at a time step k are defined by Eq. (3.10).

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \tag{3.10}$$

Unfortunately, all real systems have an inherent noise, the process noise $\mathbf{w}_k$, and any estimate based only on Eq. (3.10) will become progressively worse at each iteration. The Kalman filter assumes that all noise comes from a zero mean, multivariate Gaussian distribution. Therefore, the process noise in Eq. (3.10) is defined as $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$, where $\mathbf{Q}_k$ is the covariance of the process noise.

Fortunately, we can measure the results of the states, i.e., the system behavior, and with the *measurement model* $\mathbf{H}_k$ of the system, we can model the measurements $\mathbf{z}_k$, Eq. (3.11).

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{3.11}$$

Due to the imperfection of the sensors, all measurements contain some error due to the measurement noise $\mathbf{v}_k$. This noise is also drawn from the multivariate Gaussian distribution with zero mean, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$, where $\mathbf{R}_k$ is the covariance of the measurement noise.

The iterative estimation of states estimated by Eq. (3.10) will have a degrading performance. On the other hand, the measurement model in Eq. (3.11) and the real measurements can improve the estimation of the system states. In the Kalman filter, the estimate of the new state is recursively computed in two steps: the prediction step and the update step.

In the prediction step, we use Eq. (3.10) to obtain the expected, i.e., predicted state estimate $\mathbf{x}_{k|k-1}$. The estimate $\mathbf{x}_{k|k-1}$ is also called the *a priori* estimate because it is computed for time step k before the measurements from time step k. A "$k|k-1$" index means that the estimate for time step k is based only on the state estimate from time step $k-1$.

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k \tag{3.12}$$

In addition to the state prediction, the uncertainty, i.e., the variance, of the system states in the next step is also predicted, Eq. (3.13).

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k \tag{3.13}$$

The update step is initiated with the arrival of the new measurements. First, we compute the difference between the actual measurements $\mathbf{z}_k$ and the expected measurements, Eq. (3.14). This difference is called the *innovation*.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k|k-1} \tag{3.14}$$

Since both the predicted states $\mathbf{x}_{k|k-1}$ and the measurements $\mathbf{z}_k$ are subject to some uncertainty, the innovation should also be subject to uncertainty, and its variance is calculated using Eq. (3.15).

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{3.15}$$

Now that we know the innovation, the predicted estimate of the system states is corrected using Eq. (3.16).

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{3.16}$$

The updated state estimate $\mathbf{x}_{k|k}$ is also called *a posteriori* because it is computed once the measurements at time step k are available. The matrix $\mathbf{K}_k$ is called the Kalman gain and it decides whether the updated estimate should depend more on the predicted states or on the new measurements. There are examples where the Kalman gain has a constant value, but for the optimality of the Kalman filter, the gain must be calculated using Eq. (3.17).

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{3.17}$$

At the end, the variance of the state estimate is also updated, Eq. (3.18), and the new state estimate $\mathbf{x}_{k|k}$ and its variance $\mathbf{P}_{k|k}$ are ready for the next cycle.

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_{k|k-1} \tag{3.18}$$

The assumptions of the Kalman filter that the process and measurement models are linear limit its use cases. Therefore, in most real-world applications, the Kalman filter is adapted to the existing nonlinearities by linearizing the process and measurement models around the estimated states. This modification of the Kalman filter is called the Extended Kalman Filter.

Reduced space SGM

Initially, no prior information is available and the SGM is performed on the entire search space, but all other steps have the predicted disparity map $D_{k|k-1}$ and variance $P_{k|k-1}$. The measured disparity of a pixel $\mathbf{p}$, i.e., the disparity computed with SGM, is expected to be in the range of disparity $D_{k|k-1}(\mathbf{p}) \pm P_{k|k-1}(\mathbf{p})$.

MATCHING COST. The first step of the SGM in the reduced space is a computation of matching cost using one of the presented similarity measures. Since the KITTI dataset is chosen for the evaluation of the method, and it has sequences recorded in light-challenging outdoor environments, we chose to use the similarity cost of the census transform since it was shown to be robust under the described conditions [85]. The census transform is presented in Section 2.1.3.1, and for real-time applications, a 5×5 pixel window has shown to perform well in terms of both runtime and accuracy [86].

The complexity of the matching cost step per image is $\mathcal{O}(h \times w \times d_{\text{range}})$, where h and w are the image dimensions and d_{range} is the disparity search range. The value of d_{range} depends on the maximum disparity value expected in the scene, typically between 16 and 256. This computationally intensive step is sped up by using the SEE instruction set, which allows simultaneous processing of the census transform for 16 pixels in an 8-bit image using 128-bit SEE registers. The actual speed increase is less than 16 due to the additional overhead in preparing the data alignment for the SEE registers. After census transform, the distance between binary words is calculated using the `xor` operation supported by SSE. The matching cost computation step is parallelized by using multiple threads on horizontal strips of the image, since the calculation is independent for each pixel.

↬ AGGREGATION STEP. The complexity of the loss aggregation step of the SGM with n_r aggregation directions is $\mathcal{O}(n_r \times h \times w \times d_{\text{range}})$. This is the most time-critical step in the algorithm and its implementation is done with the AVX instruction set, which uses 256-bit registers. The aggregation formula, Eq. (3.19), is a modified version of Eq. (3.5), which ensures that the cumulative loss $L_{\mathbf{r}}$ fits into 16-bit registers.

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k) + \min \begin{cases} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d) \\ L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d \pm 1) + P_1 \\ \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + i) + P_2 \end{cases} \tag{3.19}$$

Subtracting the constant value from all previous losses prevents the loss level from becoming too high, and the final disparity is not changed. The maximum loss value for a pixel that must fit in a 16-bit register is $C_{\text{max}} + P_2$. Since the loss is written to 16-bit registers, the AVX instruction set can process 16 disparity values simultaneously. Since 16 disparity values are processed simultaneously, it does not make sense to reduce the disparity search space further.

↬ DISPARITY COMPUTATION AND REFINEMENT. The final steps are the same as in the original SGM implementation. The disparity values are determined using the winner-takes-all principle, and the parallelization is done in the same way as in matching cost computation step. The remaining noise in the disparity map is reduced using a 3×3 median filter, which also fills the values of the invalid pixels in the disparity map. This refinement is performed in the post-processing block outside the Kalman filter framework, Fig. 3.3, and does not affect the recursive filtering process.

Disparity estimation with the Kalman filter

The reduced search space is based on the estimates of the disparity mean $D_{k|k-1}$ and the variance $P_{k|k-1}$, which are recursively estimated using the Kalman filter. The pixel-wise Kalman filter independently estimates the state of each pixel in the disparity map, i.e., $w \times h$ Kalman filters are used. The filtering is divided into: (i) prediction, where we transform the information from the previous step to the current step, (ii) measurement, which was

described in the previous subsection, and (iii) correction, where the information from prediction and measurement is merged.

PREDICTION STEP. In prediction, the information about the disparity map D_{k-1} and variance P_{k-1} from the previous step and the camera motion between steps $k-1$ and k computed by the visual odometry algorithm are used to obtain the wrapped disparity map $D_{k|k-1}$ and variance $P_{k|k-1}$. As proposed in [87], the filter is applied directly in the disparity state-space. Based on the equation for pinhole projection, Eq. (2.1), a projected 3D point $\mathbf{X}_C = (X_C, Y_C, Z_C)^T$ in the camera coordinate frame is obtained with the equations

$$u_c = f\frac{X_C}{Z_C} \tag{3.20}$$

$$v_c = f\frac{Y_C}{Z_C} \tag{3.21}$$

$$d = u_c^{\text{left}} - u_c^{\text{right}} = f\frac{X_C}{Z_C} - f\frac{X_C - B}{Z_C} = f\frac{B}{Z_C} \tag{3.22}$$

where the baseline B is the displacement of the right camera and both cameras have the same focal length f. Note that the coordinates u_c and v_c are in the camera coordinate frame, so no principal point offset is required. Equations (3.20)-(3.22) convert Euclidean space coordinates to disparity space coordinates. The matrix formulation in homogeneous coordinates is given in Eq. (3.23).

$$Z_C\begin{bmatrix} u_c \\ v_c \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} fX_C \\ fY_C \\ fB \\ Z_C \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & fB & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_C \\ Y_C \\ 1 \\ Z_C \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \tag{3.23}$$

If we denote coordinates in disparity space with ω and most right matrix in Eq. (3.23) with Γ, we can write

$$\begin{bmatrix} \omega \\ 1 \end{bmatrix} = \Gamma\begin{bmatrix} \mathbf{X}_C \\ 1 \end{bmatrix} \tag{3.24}$$

The matrix Γ transforming Euclidean space into disparity space is used to express the transformation of the visual odometry pose T_{k-1}^k in disparity space H_{k-1}^k.

$$H_{k-1}^k = \Gamma T_{k-1}^k \Gamma^{-1} \tag{3.25}$$

The prediction of the disparity map $D_{k|k-1}$ is computed per pixel using Eq. (3.26).

$$\begin{bmatrix} \omega_{k|k-1} \\ 1 \end{bmatrix} = H_{k-1}^k\begin{bmatrix} \omega_{k-1} \\ 1 \end{bmatrix} \tag{3.26}$$

The prediction step of the Kalman filter also estimates the predicted variance of the disparity map $P_{k|k-1}$. Although ω has three coordinates, we do not consider the statistical properties of the first two coordinates, and the variance prediction is computed only for the disparity value. The prediction of variance in the Kalman filter for scalars is given with

$$P_{k|k-1}(\mathbf{p}) = (\Phi_{k-1}(\mathbf{p}))^2 P_{k-1}(\mathbf{p}) + q_k \tag{3.27}$$

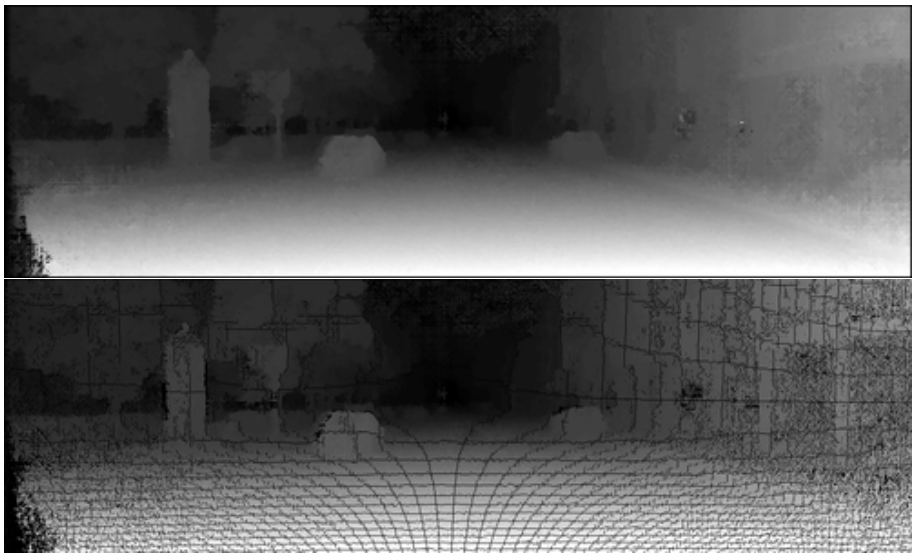


Figure 3.4: Top image: A computed disparity map D_{k-1} with 135th KITTI *Stereo 2015* benchmark pair. Bottom image: prediction $D_{k|k-1}$ based on visual odometry pose transformation. Some pixels are left without the predicted disparity because no pixel from the previous step maps to them, causing the zooming effect.

where Φ_k is a process model and q_k is process noise. The variance is transformed from the previous step using the process model and additional noise is added to model the uncertainty resulting from the computed pose transformation. The estimation of the process noise is described in Section 3.3.

Since the process model is not available in analytical form and the predicted disparity value has already been computed, the authors in [78] proposed the following approximation

$$\Phi_{k-1}(\mathbf{p}) = \frac{D_{k|k-1}(\mathbf{p})}{D_{k-1}(\mathbf{p})} \tag{3.28}$$

This approximation avoids the analytical expression for the process model. Forward camera movements lead to holes in the predicted disparity map (Fig 3.4). This *zooming effect* is a consequence of warping, where not all pixels in the new disparity map were mapped one-to-one. To correct for the missing disparities, new disparity and variance values are defined for each pixel with either 2 vertical or 2 horizontal neighbors with known disparity. For these pixels, the disparity and variance values are defined as the average of their neighbors.

↬ CORRECTION STEP. The area of the disparity search space is defined with the predicted disparity map and variance $(D_{k|k-1}(\mathbf{p}) \pm P_{k|k-1}(\mathbf{p}))$. Once SGM has computed the disparity map with the reduced disparity search space D_k^z, this is a new measurement in the Kalman filter framework. The new measurement is merged with the prediction estimate in the correction step, given in Eq. (3.29) and Eq. (3.30).

$$D_k(\mathbf{p}) = D_{k|k-1}(\mathbf{p}) + K_k\left(D_k^z(\mathbf{p}) - D_{k|k-1}(\mathbf{p})\right) \tag{3.29}$$

$$P_k(\mathbf{p}) = \left(1 - K_k(\mathbf{p})\right)^2 P_{k|k-1}(\mathbf{p}) + K_k^2 r_k \tag{3.30}$$

Here, K_k is the Kalman gain, Eq. (3.31), and r_k is the measurement noise. The measurement noise, the estimation of which is described in the following section, provides information about the uncertainty of the SGM method.

$$K_k(\mathbf{p}) = \frac{P(\mathbf{p})_{k|k-1}}{P(\mathbf{p})_{k|k-1} + r_k} \tag{3.31}$$

ESTIMATION OF PROCESS AND MEASUREMENT NOISE

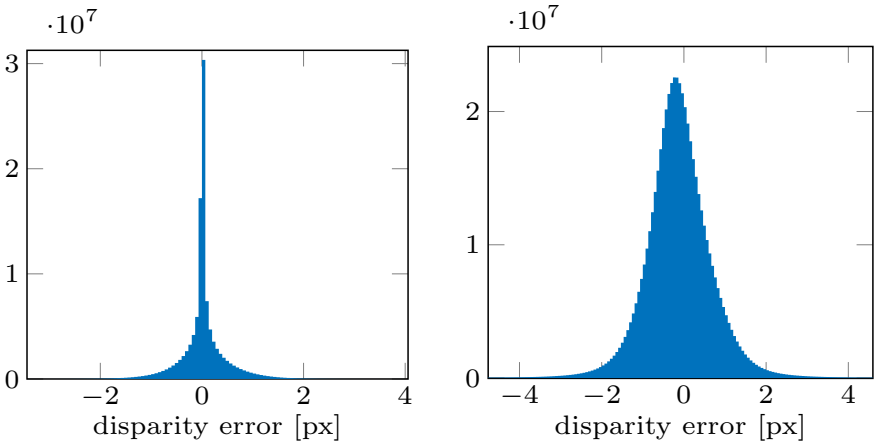
The reduced disparity search space depends on the process and measurement noise, and their accurate estimation is important for accurately eliminating unimportant disparity values. These values are estimated from the ground truth data of the KITTI dataset. The process noise estimate is the estimate of the uncertainty of visual odometry in the disparity space. In visual odometry, the moving objects are random errors that can unnecessarily increase the process noise. Therefore, process and measurement noise were estimated on 7000 images of scenes without moving objects extracted from 6 different sequences.

Ego-motion was computed using the open-source libviso2 library [88]. To filter out the error introduced by the ego-motion estimation, we used only parts of the sequences where the only motion in the frames was from the moving camera. Ego-motion estimates the pose transformation between two consecutive stereo pair images $k-1$ and k. The computed ego-motion was used to transfer the ground truth measurements obtained with a 3D laser sensor from frame $k-1$ to frame k. The transferred measurements were then compared to the ground truth of frame k. We assumed that the differences between the transferred measurements of $k-1$ and the measurements of k came from the errors in ego-motion estimation and that their variance is the process noise of the Kalman filter. Figure 3.5a shows the histogram of the disparity error in pixels, which shows that more than 99% of the errors are within ±1 pixel. The error of the ego-motion estimation accumulated over one of the evaluation sequences is shown in Fig. 3.6.

The measurement noise is the error in the disparity map computed using the proposed implementation of the SGM algorithm. The estimation is based only on the parts of the sequences without moving objects, which were also used for the process noise estimation. The left and right intensity images were used along with the 3D laser range sensor measurements from the single time step k to compute and evaluate the disparity. For each time step k, the SGM implementation computed the disparity without reducing the search space to avoid possible additional errors caused by searching for the disparity in the wrong region. In other words, each image pair was processed as if it had appeared when the Kalman filter was initialized. Figure 3.5b shows the measurements of the error values for one of the image sequences.

MOVING OBJECT DETECTION WITH A DYNAMIC STEREO CAMERA

Visual odometry assumes that all motion in the scene between time steps $k-1$ and k originates from the moving camera. In the case of a moving object in the scene, with respect to the ground plane, two scenarios are possible: (i) the moving object occupies a small portion of the scene and is filtered out of the visual odometry estimate; (ii) the moving



(a) Process noise (b) Measurement noise

Figure 3.5: Histograms of disparity errors accumulated over the training sequences. The histograms are used to determine the process and measurement noise.

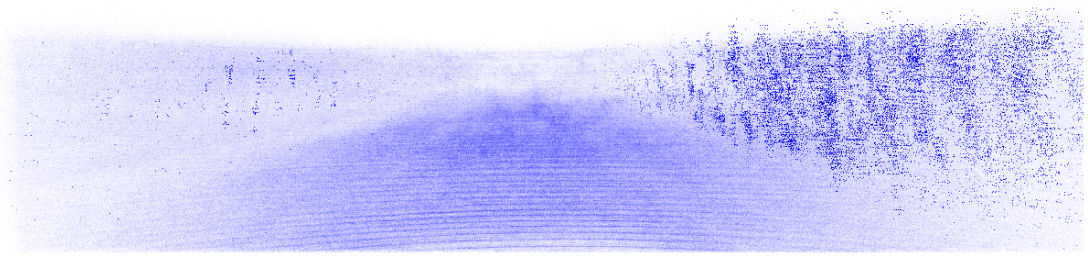


Figure 3.6: The visualization of disparity error accumulated over sequence 61. The intensity of each pixel represents the amount of the accumulated error.

object is dominant in the scene and degrades the estimate. The latter scenario often deceives drivers waiting at traffic lights when the movements of vehicles in adjacent lanes create the illusion that they are moving backward. Here we focus on the scenario with non-dominant moving objects, where the ego-motion is correctly estimated. Using the transformation T_{k-1}^{k}, the disparity from time step $k-1$ can be used to predict the disparity at moment k. Now, at moment k, we again compute the disparity map and compare it to the predicted disparity map. If there are no errors in the disparity and ego-motion estimates, all differences between the predicted and measured disparity maps come from changes in the scene caused by the moving object. Our goal was to obtain bounding boxes around the moving objects by detecting areas where the predicted and estimated disparity maps do not match.

For moving object detection, we define a new matrix *diff* that shows how different prediction and measurement are. Possible values of the matrix are given in Eq. (3.32), and it has been found experimentally that performance is better when constant values C_1 and C_2 ($C_2 > C_1$) are used for the disparity differences Δd_1 and Δd_2 instead of values proportional

Table 3.1: Sizes of sliding windows and corresponding thresholds. The threshold is a multiplication of window area and constant t which decreases from 1 in steps of 5%. The windows with diff value higher than the threshold are considered as potentially containing a moving object.

Window size	Threshold
20 × 20	400
30 × 30	855
40 × 60	2160
50 × 75	3187,5

to the difference.

$$diff(\mathbf{p}) = \begin{cases} C_1, |D_{k|k-1}(\mathbf{p}) - D_k(\mathbf{p})| > \Delta d_1 \\ C_2, |D_{k|k-1}(\mathbf{p}) - D_k(\mathbf{p})| > \Delta d_2 \\ 0, \quad \text{otherwise} \end{cases} \tag{3.32}$$

Pixel-wise detection of moving objects would be error-prone due to the SGM and ego-motion estimation noises. Therefore, we expect the presence of moving objects in discretized blocks of the *diff* matrix that have a high value of the sum over elements relative to the block area. The complexity of finding the sum from a block is proportional to the size of the block, but with preprocessing the complexity can become constant. In the preprocessing step, we calculate the matrix *sum* so that $sum(\mathbf{p})$ is the sum of all elements at the top left of $\mathbf{p}$. With the *sum* matrix, the sum of the blocks in *diff* defined with upper left coordinate $\mathbf{p}_1 = (x_1, y_1)$ and lower right coordinate $\mathbf{p}_2 = (x_2, y_2)$ is computed in constant time using the inclusion-exclusion principle, Eq. (3.33).

$$\sum_{x=x_1}^{x_2} \sum_{y=y1}^{y_2} \text{sum}(x, y) = \text{diff}(x_2, y_2) - \text{diff}(x_1, y_2) - \text{diff}(x_2, y_1) + \text{diff}(x_1, y_1) \tag{3.33}$$

Using the sliding window approach, whose size ranges from 20 × 20 to 50 × 75 pixels, we detect small areas in *diff* with high values. Table 3.1 shows the used window sizes and the corresponding thresholds that classify windows as potentially containing or not containing the moving object. The small sliding windows produced multiple bounding boxes for a moving object, and these bounding boxes had many overlapping regions since the sliding step was 10 pixels in both vertical and horizontal directions. Therefore, we grouped the detected bounding boxes using a greedy algorithm. The algorithm groups a pair of bounding boxes with the highest intersection-to-union ratio into the smallest bounding box containing both of them. This process is repeated until there are no more overlapping bounding boxes. Finally, an additional filter is applied to remove bounding boxes that are unlikely to contain a moving object due to their size. These bounding boxes have either width or height of fewer than 50 pixels or an area of fewer than 2500 pixels.

The moving objects in the KITTI dataset are mainly cars, pedestrians, and cyclists. Based on their relative mode of motion, we can divide them into two groups: (i) the objects moving toward or away from the camera, and (ii) the objects moving perpendicular to the camera. Detection of objects in the first group is possible when they are close to the camera and the difference between the predicted and measured disparity becomes apparent, Fig. 3.7a.

The second group of objects is easier to detect because they make changes in disparity in subsequent time steps by obscuring and uncovering parts of the background behind them, which often show a sufficient change in disparity, Fig. 3.7b. In the second group, drawing a bounding box around the large objects becomes a difficult task because the disparity changes at the object edges, but between the edges the disparity remains as predicted. This results in a moving object being split into two bounding boxes, Fig. 3.7c.

It is important to note that the presented method detects motion through two consecutive frames, as shown in the images in Fig. 3.7, which makes the bounding boxes larger than the actual moving objects in some cases. In addition, the sequences of the KITTI dataset usually have a sky in the upper third of the images, where we do not expect moving objects, so this region of the image is excluded from the detection of moving objects. This can be seen in Fig. 3.7c, where the upper part of the top image contains high values coming from the hard disparity estimation in textureless areas.

EVALUATION

The implementation of the proposed method is evaluated on the KITTI dataset and compared with the OpenCV's SGM implementation and the CNN method for disparity estimation LEAStereo [89]. As explained in Section 2.3.1, we considered it inappropriate to evaluate the disparity maps on one of the official benchmarks due to the specific requirements of the implemented algorithm. Instead, we used raw data sequences that provide ground truth (i.e., laser measurements) for multiple images per scene.

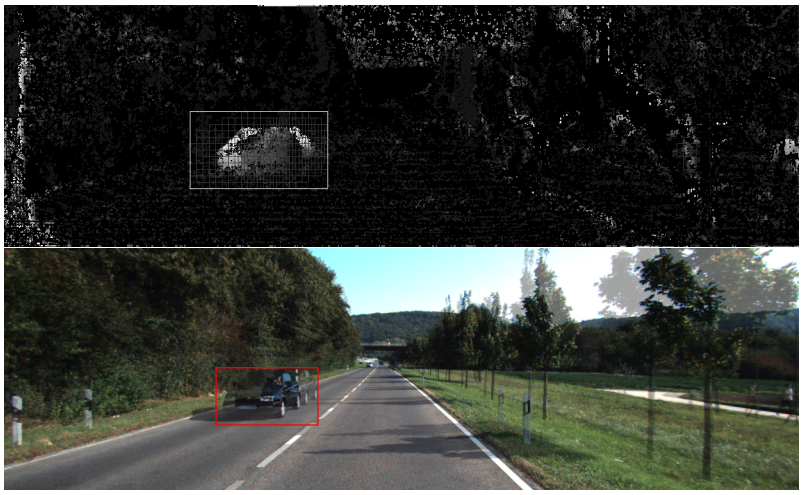
In [72] authors extended the KITTI dataset with annotations for moving objects and named it KITTI MOD. The dataset provides bounding boxes for vehicles and classifies them as static or dynamic. During the evaluation, the static detections are filtered out, leaving only the detections of moving vehicles.

Comparison of proposed and OpenCV SGM implementation

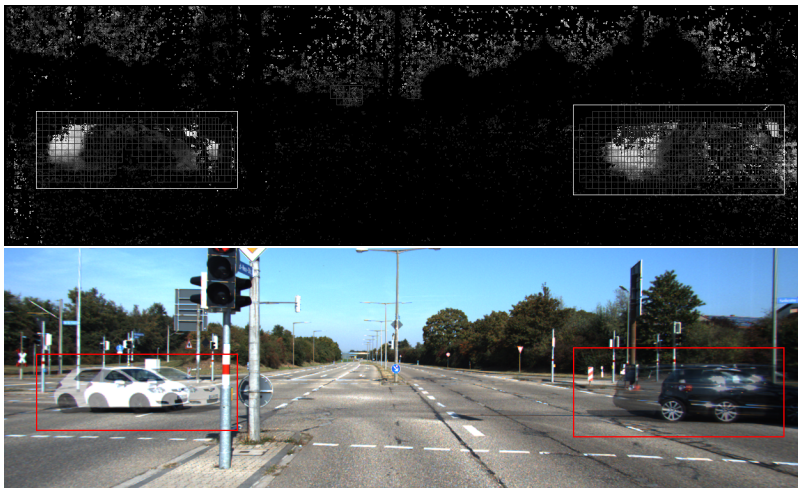
The implementation of the proposed algorithm is compared with the open source implementation of SGM from the OpenCV library. The OpenCV implementation of SGM uses the Birchfield-Thomasi similarity measure, which is robust to sampling [90], and allows the selection of 3, 4, 5, or 8 loss accumulation paths.

Prior to evaluation, we adjusted the P_1, P_2, and window size parameters for both the proposed and OpenCV SGM implementations to obtain optimal results. The parameters were tuned with a set of sequences[1] from which we removed parts with moving objects. When tuning the parameters, the proposed implementation used the entire disparity search space. The resulting parameters used for the evaluation are listed in Table 3.2, along with the values of maximum disparity and the number of loss accumulation paths used in the experiments.

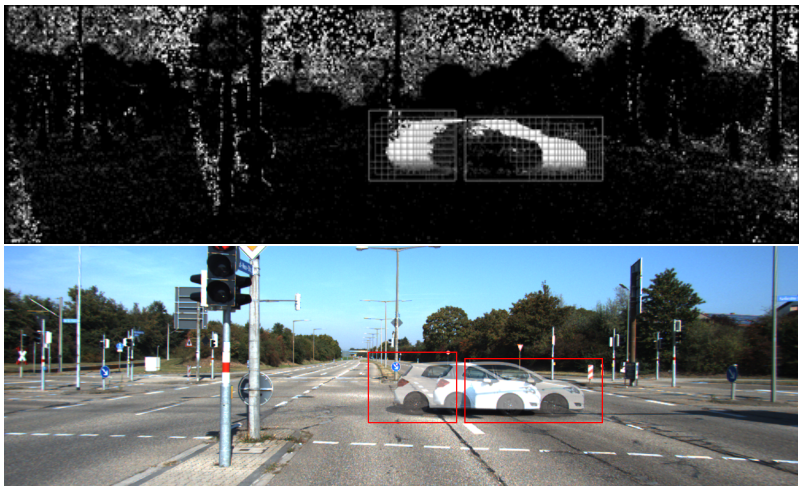
[1] The sequences used for parameters tuning: `2011_09_26_drive_0035_sync`, `2011_09_26_drive_0086_sync`, `2011_09_26_drive_0093_sync`, `2011_09_30_drive_0020_sync`, `2011_09_30_drive_0033_sync`, `2011_09_30_drive_0034_sync`



(a) Detected object moving towards camera



(b) Two objects moving perpendicular to the camera



(c) An object moving perpendicular to the camera detected with two bounding boxes.

Figure 3.7: Moving object detection on scenes 183, 45, and 46 in KITTI *scene flow 2015* benchmark. Top images in pair show detections in diff matrix image, while bottom images show detections in overlapped images from two consecutive time steps.

Table 3.2: The parameters used in the evaluation.

Parameters	OpenCV	proposed
P_1	26	6
P_2	470	65
`SADWindowSize`	3×3	-
`CensusWindowSize`	-	5×5
Loss acc. paths	8	4 & 8
Max. disp.	128	128

Table 3.3: OpenCV and proposed implementation comparison. Outliers are defined as on the KITTI benchmark (absolute threshold of 3 pixels and relative threshold of 5%). Diagonal and nondiagonal columns show the results computed using only 4 of 8 accumulations paths.

Seq.	Time per image [s]			Outliers [%]				Img. No.
	OpenCV	proposed		OpenCV	proposed			
	8-path	4-path	8-path	8-path	4-path nondiag.	4-path diag.	8-path	
19	0.25	0.17	0.19	1.07	0.58	2.11	0.83	395
39	0.26	0.17	0.19	1.33	1.09	1.85	0.97	384
51	0.28	0.16	0.18	1.72	2.41	4.49	2.72	427
61	0.28	0.18	0.19	4.64	1.96	3.74	2.11	691
84	0.26	0.16	0.18	1.27	0.75	1.70	0.76	372
96	0.26	0.17	0.19	2.38	1.35	4.18	1.65	464
117	0.26	0.17	0.19	2.93	1.54	3.88	1.75	649

The evaluation was performed with 7 sequences[2] that were not used in the parameter tuning and noise estimation. The proposed and OpenCV implementations were compared based on their accuracy and runtime. Accuracy is determined by the percentage of outliers, which is defined in the KITTI benchmark as a deviation of more than 3 pixels in absolute values and more than 5% in relative values. For the proposed implementation, we tested variants with 4 and 8 accumulation paths.

The results for the sequences are shown in Table 3.3. The implementation of the proposed method showed improved average time per frame for all sequences compared to the OpenCV implementation, with even less time per frame for the 4-path variant. In terms of accuracy, the proposed 8-path method outperformed the OpenCV implementation on 6 of 7 sequences. The scene where OpenCV had fewer outliers included parts with many moving objects in the scene. The high frequency of moving objects degrades the performance of the proposed method, which assumes a static scene. The comparison of the percentage of outliers per image pair between the SGM of OpenCV and the proposed method through the challenging sequence is shown in Fig. 3.8. The proposed method has peaks between images 200 and 380, i.e., between the images that contain a larger number of moving objects,

[2] The sequences used for evaluation: `2011_09_26_drive_0019`, `2011_09_26_drive_0039`, `2011_09_26_drive_0051`, `2011_09_26_drive_0061`, `2011_09_26_drive_0084`, `2011_09_26_drive_0096`, `2011_09_26_drive_0117`

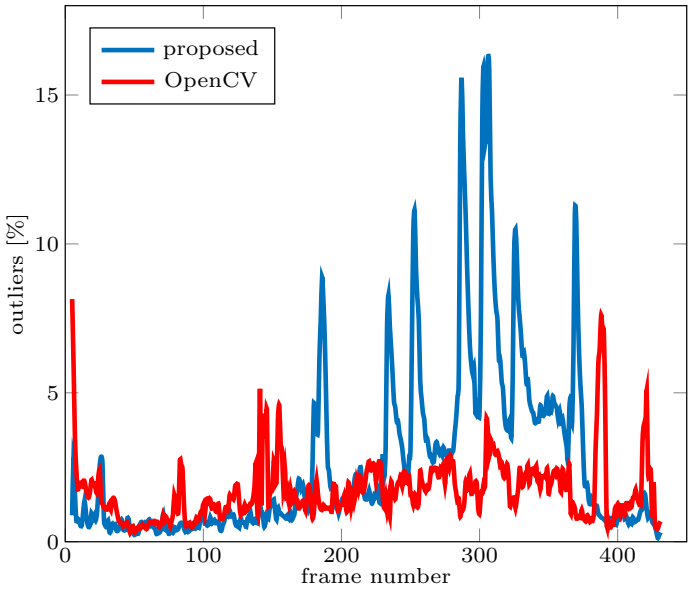


Figure 3.8: Comparison of outliers computed with OpenCV SGM and the proposed approach in the sequence *0051*. The peaks of the blue curve correspond to the frames with moving cars.



(a) Image 100 (b) Image 254

(c) Image 305 (d) Image 400

Figure 3.9: Images 100 , 254, 305, and 400 of sequence 51. The number of outliers has peaks in the images with a lot of moving objects.

Fig. 3.9.

The experiment with varying the number of loss accumulation paths yielded interesting results. As expected, the version with 4 paths takes less time than the version with 8 paths, but contrary to our expectation, the version with 4 non-diagonal paths gave slightly better results than the version with 8 paths. Further investigation showed that diagonal paths had many more outliers than non-diagonal paths. The 8-path version, which uses all paths, should have performance somewhere in between since all paths are treated the same. In [91] authors explain that the structure of the environment favors the horizontal and vertical accumulation paths over the diagonal paths. This is consistent with observations on our results.

Table 3.4: The comparison of the best performing version of the proposed SGM approach with the top-ranking learning-based approach with open source code LEAStereo. The fourth and the fifth columns show the outlier percentage. Outliers are defined as on the KITTI benchmark (absolute threshold of 3 pixels and relative threshold of 5%).

Seq.	Time per image [s]		Outliers [%]		Img. No.
	LEAStereo	4-path nondiag.	LEAStereo	4-path nondiag.	
19	0.22	0.17	0.35	0.58	395
39	0.22	0.17	0.65	1.09	384
51	0.22	0.16	0.325	2.41	427
61	0.22	0.18	0.81	1.96	691
84	0.22	0.16	0.35	0.75	372
96	0.22	0.17	0.35	1.35	464
117	0.22	0.17	0.85	1.54	649

Comparison of the proposed and learing-based approach

The proposed SGM approach is also compared to the current top ranked approach in the *KITTI Stereo 2015* benchmark with the open source implementation LEAStereo. LEAStereo is a learning-based approach that optimizes not only the neural network parameters during training, but also the architecture of the network. The network is trained in a supervised end-to-end fashion, and can generate a disparity map for the stereo image input.

The comparison between the proposed SGM approach and the LEAStereo network is shown in Table 3.4. The LEAStereo had much better performance in terms of the accuracy of the estimated disparity map and kept the percentage of outliers below 1% on all sequences. Since the network does not depend on the temporal transmission of information, the moving objects in the scene do not affect the estimation performance. This property is evident from the results where LEAStereo was able to achieve the lowest percentage of outliers exactly on the sequence with the most dynamic objects. However, the improved accuracy of LEAStereo comes at the cost of increased computational complexity. Table 3.4 shows the increased average running time per frame of LEAStereo compared to the proposed approach. Also, due to practical issues, the algorithms are run on two different platforms. The proposed approach is run on an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz and LEAStereo is run on an AMD Ryzen Threadripper 3970X 32-core processor @ 3.7GHz. It is expected that the time difference would be even more significant in favor of the proposed approach when running the test on the same platform.

Moving objects detection test

The KITTI MOD extension provides vehicle detections on the *KITTI Scene Flow* benchmark. The training image set was used to find the best parameters for moving object detection, namely C_1 (0.4), C_2 (1.5), Δd_1 (0), Δd_2 (1) and window sizes along with the thresholds given in Table 3.1. The parameters were set to maximize the F1 score, with the classification of detections determined by the intersection-over-union metric. The metric divides the overlap area of the estimated bounding box and the ground truth bounding box with their union. A true positive bounding box is the one that has an intersection-over-union value above 20%.

The proposed moving object detection method draws the bounding boxes around the area where the moving object has caused an unexpected change in disparity. Thus, the bounding boxes surround the area where the moving object is located in the past and in the present. Figure 3.10 shows images in columns of two consecutive moments, with the left and right columns representing moments $k-1$ and k, respectively. Both columns contain the detections of the proposed method, while the left column contains red bounding boxes representing the ground truth detections. The image in Fig. 3.10a does not contain ground truth detections due to the property of KITTI MOD, which only detects vehicles that are fully present in the scene. Two objects that are close to each other can be merged into a bounding box, Fig. 3.10h. And, as explained before, one object can be split into two bounding boxes, Fig. 3.10c.

The proposed moving object detection method achieved an F1 score of 59.93%, while precision and recall were 62.77% and 57.33%, respectively. In some cases, the low intersection-over-union values come from large bounding box size around the detections of the proposed method, and sometimes also from not ideal ground truth detections which do not have a bounding box around the object that is not fully present in the scene. The results show that the proposed method still needs to be improved for this challenging task of detecting moving objects with an onboard camera, for example, by forwarding the detection information to an object tracking algorithm.

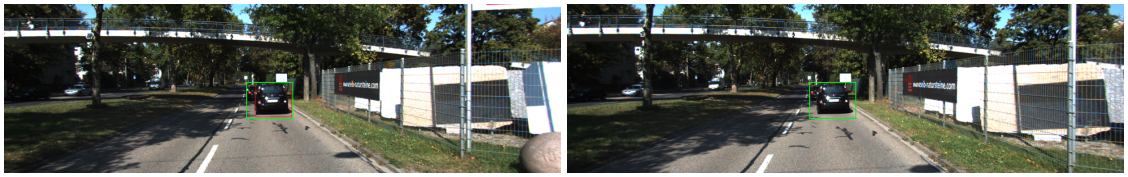
SUMMARY

This chapter describes a modification of the SGM method, which improves runtime by limiting the disparity search space and is also able to detect moving objects in the scene. After first introducing the steps of the original SGM method, we introduced the idea of passing the disparity information and fusing the information from subsequent time steps in the recursive Kalman filter framework. Estimating the disparity and variance from the Kalman filter helped to reduce the time needed to compute the new disparity. The improvement is due to the reduction in the range of possible disparity values considered in the matching cost and cost aggregation steps. In addition to the overview of all steps, we present an approach to estimate the process and measurement noise of the Kalman filter on the KITTI dataset. The process noise, which represents the uncertainty of visual odometry, is estimated by warping the ground truth of the previous time step with the ego-motion estimate and comparing the wrapped ground truth with the ground truth of the current time step. The uncertainty of the SGM, expressed as the measurement noise of the Kalman filter, is estimated by comparing its disparity map with the ground truth values. The reduced search space method is based on the assumption that the scene is static, which means that all motion in the images comes from the camera. We describe the method that detects moving objects in the areas where they have caused an unexpected change in the disparity map.

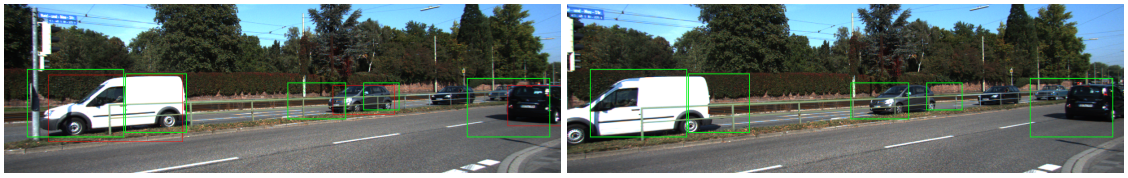
The implemented method is experimentally evaluated on the KITTI dataset and compared with the SGM implementation of OpenCV. The results show that the proposed method performs better than the OpenCV SGM in terms of both time and accuracy. The scene with a high frequency of moving objects showed that the proposed method can only be used in mostly static scenes. Moreover, an experiment with variations in the number of loss



(a) Scene 17



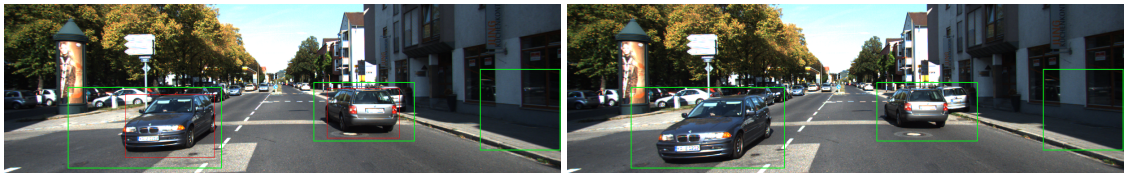
(b) Scene 21



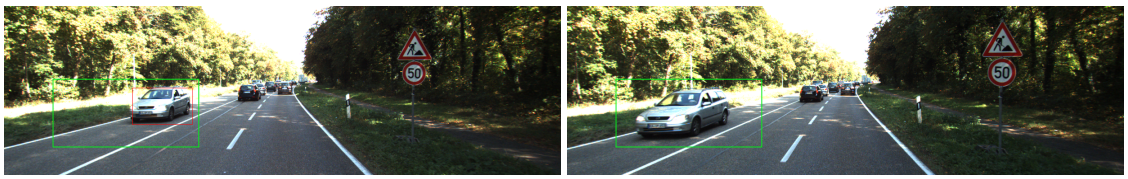
(c) Scene 23



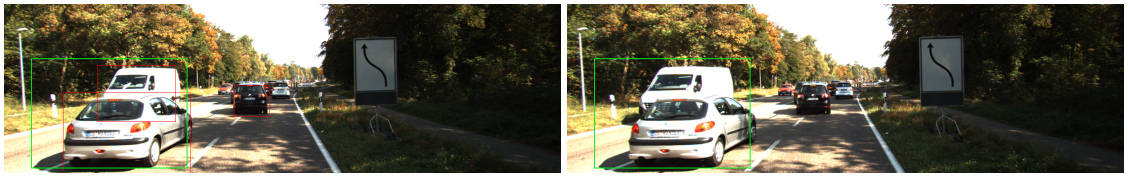
(d) Scene 45



(e) Scene 137



(f) Scene 158



(g) Scene 163



(h) Scene 180

Figure 3.10: Images from two consecutive time steps from KITTI sceneflow benchmark.

accumulation paths showed that the structure of the environment favors some directions over others.

The comparison with the learning-based approach LEAStereo showed that the proposed method is not the best option for applications focused only on the quality of disparity. On the other hand, the runtime comparison showed that LEAStereo is not suitable for real-time applications with limited resources.

The moving object detection approach tested on the KITTI MOD dataset, showed that it is possible to detect a low frequency of moving objects in the scene. The use case limitations of the approach, as well as the low F1 score on the dataset, indicate that the presented approach is not able to produce reliable results for this challenging task and that additional post-processing is required.

4

Stereo visual localization of autonomous agents in robotized warehouses

The modern supply chain is a highly complex system that needs to be automated at many levels. One of the components of this complex system is warehouses, where raw materials, product parts or finished goods are stored. The urgent need to increase supply chain throughput, improve warehouse performance and reduce costs has brought warehouse automation solutions into the focus of the logistics community. Automation solutions include warehouse management systems (WMS), automated guided vehicles (AGVs), fleet management systems (FMS), and automated storage and retrieval systems. Implementing automation solutions improves warehouse efficiency by removing difficult and repetitive tasks from human workers. Also, a different paradigm where robots carry racks to human workers can double productivity and increase warehouse flexibility [92].

Another important aspect of warehouse automation is the safety of human workers. Safety in automated warehouses is usually ensured by a safety fence that separates the automated area and the human worker area. This effective solution guarantees the safety of workers because as soon as the safety fence is breached, all robots are stopped. This happens when a human worker enters the manufacturing area to perform a task that the robots cannot do, such as repairing a broken robot, picking tasks, or lifting a dropped product. Frequent breaches of such safety systems can lead to an efficiency deficit in large warehouses with large fleets of robots. Therefore, warehouses with large fleets would benefit from a different type of safety system that is self-adjusting and stops only those robots that are potentially dangerous to the worker. Solutions such as section-by-section locks that stop robots in specific areas improve flexibility in maintenance situations where a human worker remains in place while the task is being performed. On the other hand, sectional locks are difficult to apply to picking tasks because the worker is free to move around the warehouse.

In addition to increased efficiency, the safety system that allows humans and robots to work simultaneously in the same area also offers the potential for increased flexibility through human-robot collaboration. The authors in [93] provide an overview of the safety mechanisms used for human-robot collaboration in manufacturing processes. However, warehouses do not require direct human-robot collaboration and a safety system that allows coexistence would be ideal for warehouse use cases. Such a system would only stop the robots when they are near the worker. To maximize warehouse efficiency, the safety system would also track the location of the workers and redirect the robots so that no stoppage

occurs.

In the remainder of this chapter, we first give an overview of the Safelog project safety levels that motivated the development of a localization method for automated warehouses. We then introduce the components of the localization approach: (i) localization using fiducial markers and (ii) visual odometry. Then, we describe the fusion of these two location cues into a globally correct location estimate in the graph optimization framework. Later, the proposed localization method is evaluated on self-recorded datasets and compared with the state-of-the-art SLAM method ORB-SLAM2. At the end of the chapter, we provide a summary of the proposed method and the evaluation results.

MOTIVATION

The SafeLog project is a Horizon 2020 project [5] with the goal of developing a safety concept for human-robot interaction in automated warehouses. The safety concept is divided into the following levels: (i) safety level C, which prevents human-robot encounters by optimally routing robots and humans; (ii) safety level B, which warns humans and robots if the encounter cannot be avoided; and (iii) safety level A, which shuts down the robots for which robot rerouting was not possible and which therefore came close to humans. Safety level A has strict reliability requirements, as it is the last line to ensure safety. It is implemented with ultra-wideband (UWB) sensors attached to both robots and humans. The UWB sensor modules continuously measure the distance between each other. When this distance falls below a predefined threshold, the robot carrying the UWB module is stopped. Less stringent safety requirements, on the other hand, are placed on safety levels B and C. To implement safety levels B and C, the WMS requires correct positions and path plans of all people and robots in the warehouse. The path plans of the robots can be easily obtained from the FMS, while the movements of the humans must be estimated based on their tasks and behaviors [94]. Also, safety levels B and C depend heavily on the information about the position of the workers, which is obtained through the localization process.

Localization in warehouse environments has mainly been focused on automated robots and products, and there is a spectrum of localization systems aimed at solving this problem. In [95] authors divide localization technologies into the following categories: UWB, Radio Frequency Identification (RFID) systems, vision systems, and Wi-Fi technology. Accurate localization of autonomous forklifts is successfully solved using the UWB and RFID technologies. Although they provide accurate localization, the areas of high localization accuracy of such systems are limited to picking or delivery stations, where the accurate positioning is of great importance. For the localization of humans, a system that provides good accuracy for the entire area of the warehouse is needed. Visual systems such as those in [96] and [97] perform localization by placing a series of cameras in the environment that track objects within their field of view. Such a localization system can cover a large area with sufficient accuracy, but covering the entire warehouse would require many camera sensors, and the installation and calibration of such a system would be time consuming. Another approach is proposed in [98] and [99], where localization is performed using a set of wearable sensors, such as 2D laser sensors, camera sensors, and IMUs mounted on a backpack. The lasers can provide accurate measurements for a sparse set of points, which

could improve localization, but their weight would become cumbersome after a long time when carried by a human worker. Therefore, we decided to use a set of small, lightweight, and low power consumption sensors that can be worn for a long period of time without affecting human working conditions. The visual camera sensors meet the size and weight requirements and are also low cost and informative [100]. Moreover, it is possible to use multiple cameras in the same environment because they are passive and do not interfere with each other.

The automated warehouses have a predefined structure that can be used in localization. In [101] authors used the environment structure to localize a mobile robot. In that work, a mobile robot used wheeled odometry and fiducial ground marker detection to estimate its location in a warehouse-like environment. Our proposed method is similar to this work, but instead of using wheeled odometry, which limits the application to the wheeled robots, we use visual odometry, which has higher accuracy than wheeled odometry [102].

The state-of-the-art SLAM solutions such as [53], [58], [70] perform localization based solely on wearable sensors such as IMU and cameras. The solutions showed impressive results on public datasets such as KITTI and EuRoC [103] and one could assume that one of these solutions could be used for the localization of people in warehouses. However, localization in warehouses is specific for several reasons. First, the visual SLAM solutions create the environment map assuming that the environment is static. However, an automated warehouse with a fleet of robots moving shelves does not meet the assumption of a static environment. Second, warehouse environments are places with a highly present visual aliasing effect because many places look similar to each other due to the uniformly distributed and equally sized shelves. The visual aliasing increases the risk of incorrect loop closure, which would affect the localization performance. Third, localizing a human with a wearable system limits the size and weight of the processing equipment. With the frequent loop closure expected in warehouse localization, the limited processing power might be insufficient for an intensive SLAM approach in real-time. Therefore, the proposed solution for worker localization must be lightweight and based on the detection of stable features in the environment. In automated warehouses, it is possible to use a set of unique floor markers with known poses that are used for robot localization. The map of these markers is static and easy to process. Therefore, we decided to use them to obtain a unique, globally correct human pose. The use of this map does not require any additional effort in the setup phase, since the markers are used for localizing the robots and are already available.

An overview of the proposed visual system for human localization is given in Fig. 4.1. Initialization begins with the detection of the initial ground marker at the warehouse entrance in the image from the downward-looking camera. Initialization determines the initial transformation between the camera sensor suite and the warehouse coordinate system. From then on, the relative pose is computed with visual odometry based on the stereo camera input. Each time a new marker is detected, the estimated absolute pose is fused with other location information. The following two sections describe the methods used to estimate global and relative pose.

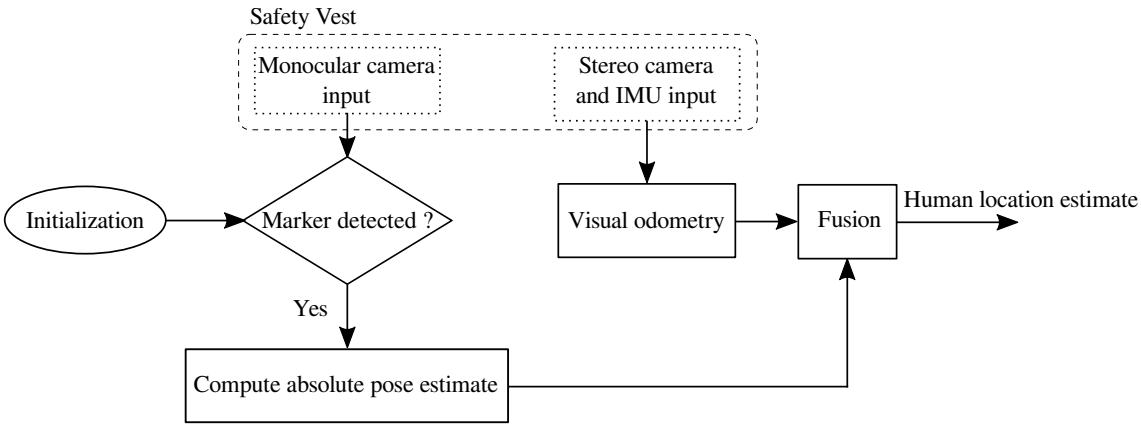


Figure 4.1: The concept of the proposed visual human localization system.

GLOBALLY CORRECT VISUAL LOCALIZATION WITH FIDUCIAL MARKERS

The automated warehouses with a fleet of autonomous robots localize them with a set of fiducial ground markers sparsely distributed throughout the warehouse. The robots have visual sensors on their undersides that detect and identify the markers as they cross them. The markers are unique, and the warehouse management system has accurate poses of all the markers in the warehouse. The proposed method focuses on the markers provided by the SafeLog project partners - the company Swisslog [104], which develops products and systems for logistics automation. These markers have already been presented in Section 2.3.2 and are shown in Fig. 2.7. The marker has a size of 10×10 cm^2 with 9 squares of 1.4×1.4 cm^2. This size is sufficient for detection and identification with a ground robot whose sensors are a few centimeters from the ground. Markers of similar size are also expected in the solutions of other companies that also use markers for robot localization. On the other hand, the camera sensor unit is located on the lower back of the human, which is more than a meter from the ground, and a special method is needed for fast and accurate pose estimation. The marker detection algorithm was developed by the partners in the SafeLog project [105], but for the completeness of the localization method described in this thesis the steps of the algorithm are presented here.

The three main steps of the marker detection algorithm are:

- Detection of a ground marker and identification of the region of interest (ROI) around the ground marker
- Identification of the ground marker
- Computation of the relative camera pose

The goal of the first step is to determine if a marker is present in the image. If so, the position of the marker is located and unimportant parts of the image are removed to improve the processing time of later steps. Marker detection is performed by matching the ORB features [69] between the input image and the reference ground marker image, Fig. 4.2. After clustering the ORB matches, a cluster with the highest number of matching features is selected and the ROI around the center of the cluster is used for further processing. Each

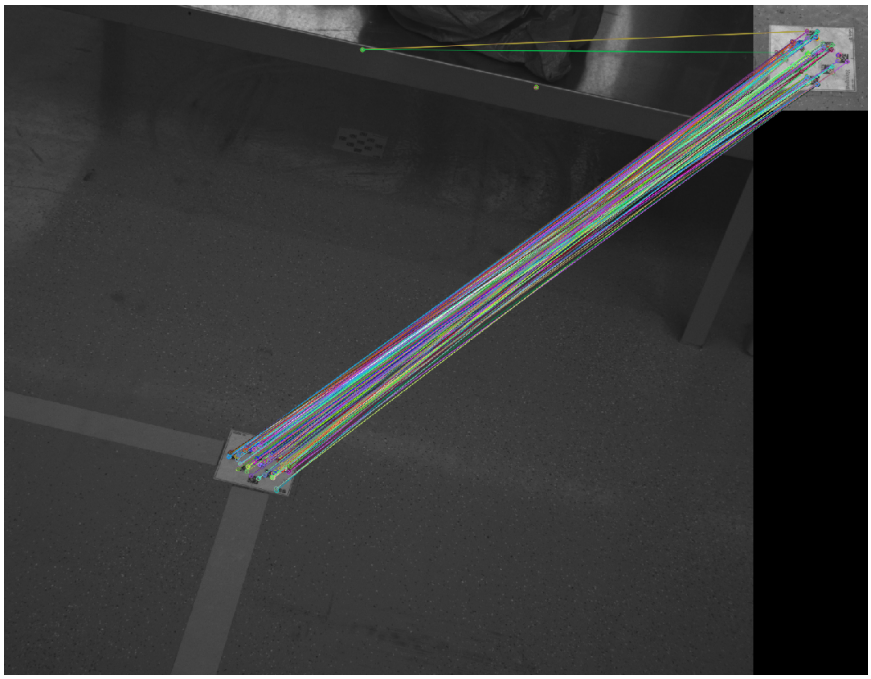


Figure 4.2: Matching ORB features in the downward-looking camera image to the reference marker image.

marker has its ID encoded in 9 squares with a two-dimensional code called DataMatrix. The marker ID is retrieved using the identification algorithm implemented in *libdmtx* [106]. Once the ID of the marker is known, its absolute pose in the warehouse reference system is determined using the marker look-up table. Then, an estimate of the relative transformation between the camera and the marker, and a known absolute marker pose are used to compute the absolute pose of the camera, i.e., the human carrying the camera.

The relative transformation between the camera and the marker is computed in the steps shown in Fig. 4.3. The initial image in this step is the input image, which is cropped to ROI and contains the marker, Fig. 4.3a. The image is blurred by the process of morphological opening which results in the image shown in Fig. 4.3b. The morphological opening darkens the DataMatrix squares and whiteness the rest of the marker, which helps the correlation process in the next step to find the DataMatrix centers. For the correlation step, the grayscale values of the cropped image and the kernel shown in Fig. 4.3f are scaled so that the gray value is 0 and the black and white values are -0.5 and 0.5, respectively. The optimal size of the kernel depends on the size of the DataMatrix, which is not known, but it has been found experimentally that sizes between 50 and 70 pixels give good results. Convolution with a kernel of this size is very computationally expensive, so the convolution is computed on the cropped image. The round shape of the kernel ensures that all orientations of markers are treated equally. The correlation with the double kernel leads to the image shown in Fig. 4.3c. After thresholding the correlated image, 9 distinct clusters appear, Fig. 4.3d. The centers of these clusters are passed to the Perspective-n-Points (PnP) method, which computes the relative transformation, as shown in Fig. 4.3e.

Fiducial ground marker detection can provide a globally correct estimate of the pose in discrete areas where the markers are in the camera's field of view. In the warehouse, the

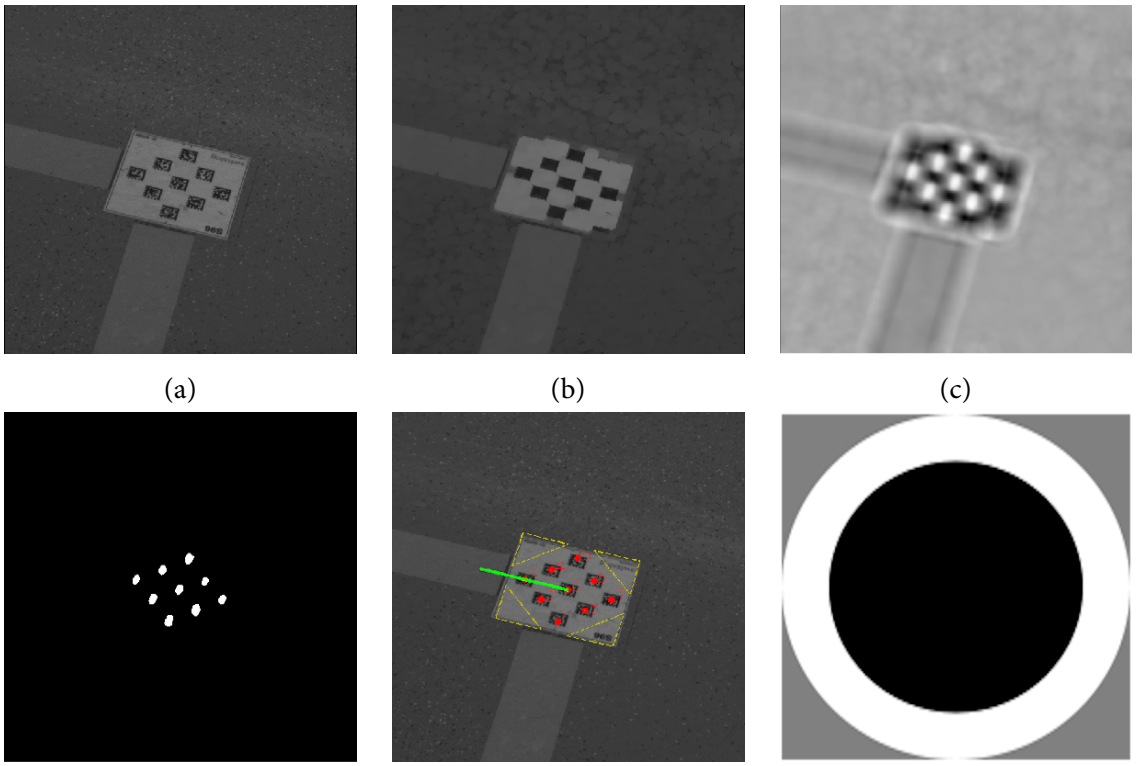


Figure 4.3: Steps of the marker-based pose estimation algorithm: a) ground marker in the original image, b) result of the morphological opening, c) image correlated with a double kernel, d) thresholded correlation image, e) marker with the computed orientation, f) the double kernel.

markers are evenly spaced at 1.2 meter intervals, which allows regular updating of the pose of the ground robot. On the other hand, the human's path may not cross the marker for a while, which means that the infrequent marker is rarely detected and consequently the pose cannot be estimated. Even if the marker is detected, interrogation of the marker's ID might be complicated by image blur, partially obscured marker, or difficult lighting conditions. In such cases, pose estimation may fail for more than half a minute, and the size of the area that the human can reach in this time, i.e., the possible poses of the human, may affect the efficiency of safe path planning in the FMS. Therefore, worker localization based on global location information must be supported by a relative pose estimation method that provides frequent and accurate pose estimates between localizations with the ground markers.

RELATIVE LOCALIZATION WITH VISUAL ODOMETRY

A frequent estimate of a human pose can be computed with visual odometry. Visual odometry computes a relative transformation of the camera given two successive pairs of images. From the sequence of images, visual odometry generates a series of transforms, and the most recent pose in the sequence is computed by concatenating the series of transforms. Implementations of visual odometry for real-time applications provide pose estimation at a constant frequency equal to the frame rate.

Visual odometry is subject to noise and the computed pose transforms contain some error rate. The error of pose estimation is accumulated by the concatenation of erroneous transformations. Even if the error is unbiased, the pose estimate may drift, as in a simple coin-flipping example of a random walk process. Therefore, visual odometry is never solely used for localization on longer sequences. In the presented use case, the constant frequency and locally accurate pose estimation of visual odometry is combined with the occasional global corrections of the ground marker pose estimation. The proposed localization solution is not limited to any particular visual odometry, and the current implementation uses the SOFT visual odometry, presented in [57], which is currently the top ranked method on the KITTI odometry benchmark [6]. The performance of the SOFT is the result of careful feature selection in the estimation process, and an overview of the method is given in the sequel.

Feature matching and tracking The first step of the SOFT method is to extract features in the image pair using the blob and corner masks introduced in [88]. Non-maximum suppression leaves a sparse set of features that are matched based on the SAD similarity measure. The outlier rejection of this fast and error-prone procedure is improved by circular matching with the image pair from the previous step. The feature matched between images I_k^l and I_k^r in step k is also matched across images $I_k^l \rightarrow I_k^r \rightarrow I_{k-1}^r \rightarrow I_{k-1}^l \rightarrow I_k^l$, and if the last feature in the matching chain matches the original one, the feature is retained. The additional check is performed using the NCC similarity measure, which is more reliable but significantly slower than the SAD.

It has been shown that the carefully selected subset of matched features improves the pose estimation of visual odometry [57]. Thus, the features are divided into groups based on their position in the image, and each group processes further feature rejection independently to ensure a spatially uniform distribution of features. In each group, the features are selected based on the following steps:

- Group the features into four classes (blob min, blob max, corner min, corner max) and sort them according to their strength
- Select the strongest feature from each class and push it to the final list until no features are left
- Use the top n features from the final list for further processing

Similar to feature grouping, where the spatial distribution of features was preserved, the temporal distribution is achieved by tracking the features. Each feature is given a unique ID, age, feature class and strength, initial descriptor, and refined position in the current image. For each new image, the age of the corresponding feature is increased by one and its position is refined with the initial descriptor. Refining the position with the initial descriptor helps to reduce the drift in pose estimate.

Finally, the features in each group are selected based on their age and class strength. It is noticed that the features that have been tracked for a longer period of time are more reliable and have a lower probability of being an outlier. Therefore, for two features, the older one

should be preferred over the other. For two features x and y, the preference is determined using the following function

$$\text{select}(x, y) = \begin{cases} \text{stronger}(x, y), & \text{if } \text{age}(x) = \text{age}(y) \\ \text{older}(x, y), & \text{if } \text{age}(x) \neq \text{age}(y) \end{cases} \tag{4.1}$$

Pose estimation The problem of pose estimation is divided into two parts: rotation estimation and translation estimation. Rotation is estimated using only the left image. Since the camera parameters are known, the epipolar constraint can be exploited, Eq. (4.2).

$$q_{k-1}^T E q_k = 0 \tag{4.2}$$

where q is a tracked feature in homogeneous coordinates at steps k and $k-1$ and E is the essential matrix containing information about the translation and rotation of the camera between two steps k and $k-1$. Equation (4.3) shows the decomposition of the essential matrix.

$$E = [t]_\times R = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.3}$$

where $[t]_\times$ is a notation for a skew symmetric matrix with elements of vector $t = (t_x, t_y, t_z)^T$. Author in [51] presents the 5-point method, which computes the essential matrix using 5 corresponding points. The RANSAC (Random Sample Consensus) method is used in combination with the 5-point method to compute the rotation estimate unaffected by the remaining outliers. The subsets of 5 features are selected from the feature set and several suggestions are made for the essential matrix. The rest of the features are used to vote for one or more proposals, and the proposal with the most votes is selected as the rotation estimate. Once the rotation is computed, the estimate can be further improved by spherical linear interpolation (SLERP).

The essential matrix contains the direction of the translation vector t, but not the scale. The second image of the stereo pair is needed to estimate the translation in the metric scale. All feature points are first triangulated in 3D using the previous stereo pair and then projected back into the camera image plane using the estimated rotation R and translation t, Eq. (4.4).

$$p = \pi(X; R, t) = \begin{bmatrix} f_x & 0 & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.4}$$

where p is a projected point in the image plane and the x, y, z coordinates of the triangulated 3D point. The translation t is computed by minimizing the following cost function over the n features in the set

$$\arg\min_t \sum_{i=1}^{n} ||p_i^l - \pi^l(X; R, t)||^2 + ||p_i^r - \pi^r(X; R, t)||^2 \tag{4.5}$$

The function is computed fast enough that each element in the feature set can suggest a solution for translation t. Given a feature set of n elements, there are n translation proposals

and for each proposal the remaining feature sets vote by being classified as outliers or inliers. The inliers of the proposal with the most votes are used in the total optimization, which is obtained from the following equation

$$\arg\min_{t} \sum_{i=1}^{n} w_i \left(||p_i^l - \pi^l (X; R, t)||^2 + ||p_i^r - \pi^r (X; R, t)||^2 \right) \tag{4.6}$$

where w_i is a feature weight computed based on the spatial and temporal properties of the feature.

IMU-aided pose estimation Some stereo cameras are equipped with an IMU that can measure the rotation of the camera. In such cases, it is possible to avoid the computationally expensive 5-point method and directly use the IMU measurements. However, the estimate of rotation from the IMU measurements is computed by accumulating the measurements. Thus, the estimate is a random walk process that can be used to estimate rotation when the accumulated error, called bias, is known. The Kalman filter is used to determine the IMU bias. It is assumed that the slowly changing bias can be correctly estimated by the series of updates coming from the image pair sequence. Between two image pairs, the IMU measurements are accumulated and result in a rotation prediction. The predicted rotation is a substitute for the rotation estimate that would otherwise be obtained using the 5-point method. In addition, the total optimization in the IMU-aided case is computed for both rotation and translation. The resulting rotation is fed back to the Kalman filter as a correction.

The details of both implementation versions can be found in [57] and [58]. The sensor setup in our use case includes a stereo camera with an integrated IMU, so the proposed localization method uses an IMU-aided SOFT version. Alternatively, in case future constraints limit the use of IMU, the purely visual version of SOFT can be used.

FUSION OF GLOBAL AND RELATIVE LOCATION CUES IN GRAPH OPTIMIZATION FRAMEWORK

So far, two sources of localization have been presented: (i) ground marker localization with globally correct infrequent pose estimates and (ii) visual odometry with a constant frequency of progressively degrading pose estimates. Localizing workers in the warehouse requires globally correct pose estimates with constant frequency, which can be achieved by merging these two localization sources. The graph optimization framework is suitable for this use case because the optimization is executed only when the new ground marker is detected. The optimization corrects the pose estimate, and between two corrections, the online pose is estimated using visual odometry.

Graph optimization

Graph optimization is the most popular approach in robotics for solving localization problems, and in recent years, state-of-the-art visual localization solutions are based on the optimization framework [107]. The main advantage of graph optimization over filter-based solutions is the ability to adaptively linearize all nonlinearities in the graph around the latest

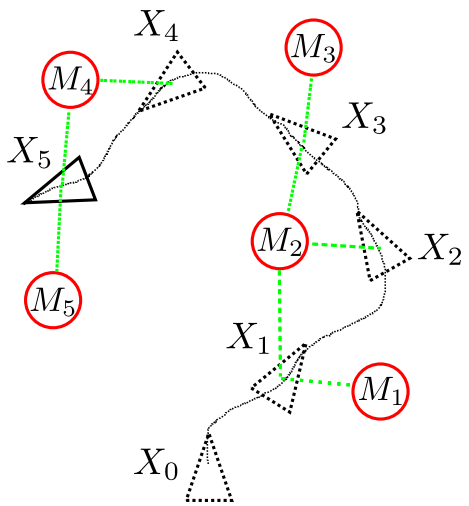


Figure 4.4: Two-dimensional visualization of an agent (black triangle) moving through an environment with features (red circles). Dashed poses are previous agent poses and the black curve connecting them represents the trajectory. When the agent is sufficiently close to a feature, sensors are able to measure the relative pose of the agent with respect to the feature (green dashed lines).

estimates of the states. This is possible by tracking all states, measurements, and map features in the graph structure and updating all graph variables simultaneously. Optimization of the graph aims to minimize the discrepancy between the estimated states and map features and the measurements.

Graph optimization in the localization problem builds a graph in the following way. The agent moving through the environment creates a trajectory $X_{0:k}$, which consists of a set of states $X_{0:k} = \{X_0, X_1, \cdots, X_k\}$ from the beginning to the current time step k, Fig. 4.4. Each state X_i is defined with 6 degrees of freedom describing the position and orientation of the robot at that time step $X_i = \{x_i, y_i, z_i, \psi_i, \gamma_i, \theta_i\}$. The map M consists of a set of discrete features $M = \{M_1, M_2, \cdots, M_n\}$, and each feature M_i has its own position and orientation. In the graph optimization framework, both the states and the map features are represented by nodes. Two nodes are connected by an edge that defines a relationship between them. Depending on the type of node the edge connects, the edge represents the relationship stemming from either the agent's motion or the agent's measurement. Two state nodes are connected with the information coming from the agent's motion, and the state and feature nodes are connected with the agent's measurement. Both types of edges are described with a mathematical model that is a function of the state and map, to which a zero-mean Gaussian noise has been added. The edge connecting two state nodes is described with the motion model, Eq. (4.7), and the edge connecting the state node and the measurement node is described with the measurement model, Eq. (4.8).

$$X_k = f(X_{k-1}, u_k) + w_k, \quad w_k \sim \mathcal{N}(0, Q_k) \tag{4.7}$$

$$z_{k,j} = h(X_k, M_j) + v_k, \quad v_k \sim \mathcal{N}(0, R_{k,j}) \tag{4.8}$$

the function f takes a previous state X_{k-1} and the control input u_k between time steps k and $k-1$ to produce a new state X_k. The function h returns the expected measurement

value $z_{k,j}$ of the map feature M_j from the state X_k. In the general case, the functions f and h are nonlinear and they are linearized during the optimization process. Both models are superpositioned with zero-mean Gaussian noises, the process noise w_k and the measurement noise v_k.

The localization problem in the graph optimization framework finds optimal states X^* and map M^* using the information from the motion and measurement models, Eq. (4.9).

$$X^*, M^* = \underset{X_{1:k}, M}{\arg\min} \sum_i \|X_i - f(X_{i-1}, u_i)\|^2_{Q_i} + \sum_{i,j} \|z_{i,j} - h(X_i, M_j)\|^2_{R_{i,j}} \tag{4.9}$$

Fusion of location cues

The graph optimization framework is adapted for the fusion use case, where the worker's pose is estimated based on information from the ground marker localization and stereo visual odometry. The nodes representing the pose of the worker and the markers are formulated as members of the SE(3) group

$$X_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix}, \qquad G_j = \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix}. \tag{4.10}$$

Each time a ground marker is detected, a marker node G_j and a human pose node X_i are added to the graph and connected with an edge N_{ij} computed by the marker detection algorithm. In addition, the human pose node X_i is connected to the previous human pose node X_{i-1} by an edge $U_{(i-1)i}$ computed by the stereo visual odometry. The edges $N_{i,j}$ and $U_{(i-1)i}$ also belong to the SE(3) group and represent the transformation from the ground marker to the current pose and the transformation from the previous pose to the current pose, respectively, Eq. (4.11).

$$X_i = U_{(i-1)i} X_{i-1}, \qquad X_i = N_{ij} G_j. \tag{4.11}$$

The warehouse provides true poses of the ground markers G, and the known map significantly reduces the complexity of the optimization, which now only optimizes the trajectory X^* of the worker. The graph optimization with the known map has the following minimization function

$$X^* = \underset{X_{1:k}}{\arg\min} \sum_i \|X_i - U_{(i-1)i} X_{i-1}\|^2_Q + \|X_i - N_{ij} G_j\|^2_R. \tag{4.12}$$

The proposed localization approach is summarized in Algorithm 1 and illustrated in Fig. 4.5 using four images. The beginning of localization process starts with the initialization procedure. The worker has to stand over the ground marker at the entrance of the warehouse. Detection of the first ground marker at the entrance initializes the localization algorithm by computing the transformation between the ground marker G_0 and the worker's pose X_0, denoted N_{00}, Fig. 4.5a. After the initialization procedure, visual odometry computes the estimate of the worker's pose as he moves through the warehouse. The detection of the next marker G_1, Fig. 4.5b, inserts two new nodes and two new edges into the graph. The new nodes are the nodes representing the pose of marker G_1 and the pose of worker X_1. The first edge N_{11}, computed with the marker detection algorithm, connects the nodes G_1 and

Algorithm 1 Proposed fusion based human localization

```
main thread:
repeat
    if marker-based pose estimation input then
        Set initial pose
until pose initialized.
Initialize graph
repeat
    if VO pose estimate then
        if marker-based pose estimates in queue then
            Create marker-odometry pair
            Push pair to the pose graph
            Set optimization flag
    if marker-based pose estimation input then
        Add marker-based pose estimate to queue
until end of the recording.
optimization thread:
repeat
    if optimization flag set then
        Optimize graph
        Return optimized graph
        Reset optimization flag
until Killed from the main thread
```

X_1, and the second edge U_{01}, which connects the previous pose X_0 to the current pose X_1, is computed with visual odometry. The visual odometry and marker detection estimates do not necessarily have the same timestamp because they are computed on images from different cameras. Therefore, each time the ground marker is detected, the visual odometry pose estimates are interpolated with the timestamps closest to the marker detection pose to obtain a pose estimate with the timestamp that matches the marker detection pose estimate. The worker's pose is continuously updated with the visual odometry, as shown by the green dashed line and circle in Fig. 4.5c. The new nodes (X_3, G_3) and edges (N_{33}, U_{13}) are added to the graph only when a new ground marker G_3 is detected (marker G_2 was not detected and therefore node X_2 was not added to the graph), as shown in Fig. 4.5d. Once the new nodes and edges are added to the graph, the optimization of the graph is started and the whole trajectory is updated, i.e., the pose nodes X_i of the workers are corrected. The g^{2}o framework [108] is used to implement the pose graph optimization. Graph optimization runs in parallel with visual odometry and ground marker detection in a separate thread, as described in Algorithm 1.

Based on the described location cues and the use case scenario, one could argue that graph optimization is not needed and the accumulated visual odometry pose error can be corrected directly with the pose estimate from the marker detection algorithm. This is partially true because the past poses in the trajectory are not important and only the

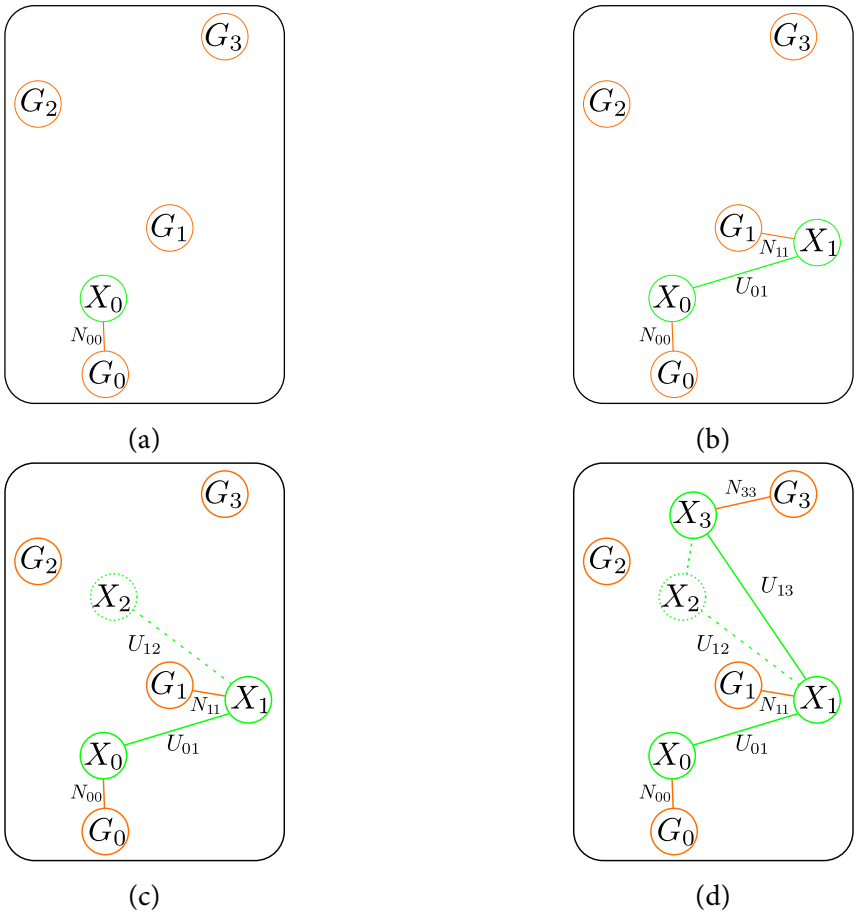


Figure 4.5: Construction of the pose graph with the pose nodes obtained from visual odometry (X_i) and the ground marker nodes (G_i). Green nodes and edges represent the output of the visual odometry algorithm, while orange nodes and edges represent the output of the marker detection algorithm.

most recent pose estimate is of interest. However, in this case, the transformation computed with the ground marker detection may have significant orientation errors, resulting in unacceptably large location errors. Therefore, we optimize the entire graph to enforce pose consistency and obtain a reliable location estimate.

EVALUATION

The proposed method for worker localization is evaluated on datasets with warehouse-like environments that have ground markers with the known pose on the floor. Based on the specific requirements, all publicly available datasets were filtered out and new customized datasets were created as described in Section 2.3.2. The proposed method, which was evaluated using the self-recorded datasets, might give the impression that the entire evaluation is done on the floating ground. Therefore, we also provide the evaluation results of one of the state-of-the-art SLAM methods, ORB-SLAM2.

The evaluation is performed using the evaluation tool[1] presented in [75]. The evaluation measure used is the *absolute trajectory error* (ATE), as commonly used for SLAM and

[1] The code used for the evaluation is available at https://github.com/uzh-rpg/rpg_trajectory_evaluation

odometry evaluation [6]. The trajectories used for the evaluation are first transformed using the Kabsch algorithm [109] to obtain the best match with the ground truth trajectory. The trajectory alignment with the Kabsch algorithm is included in the evaluation tool. All evaluations were performed on a Lenovo P51 notebook with Intel Core i7-7700HQ CPU @ 2.80GHz×8.

Dortmund dataset

The test area in the Dortmund dataset is $5 \times 3\ m^2$ and has 6 ground markers on the floor. The area is surrounded by an Optitrack motion capture system that provides almost complete coverage for recording the ground truth pose. In the recordings, one real rack is present and the others are simulated by the plastic boxes. The dataset is evaluated in two scenarios. The first one is the scenario of *standard operating conditions*, which was evaluated using 9 sequences with a static environment where the human walks through the arena and simulates the behavior of a worker in the warehouse. The second scenario is the case of a *non-static environment*, which simulates the situation of different warehouse rack layouts. This scenario is important for the warehouse use case, as the robots change the distribution of the racks as they bring them to the picking stations.

STANDARD OPERATING CONDITIONS SCENARIO. Table 4.1 shows the results of the evaluation on 9 sequences (DM01-DM09) for three trajectories: *fusion*, *odometry*, and *orbslam2*. The *fusion* trajectory is computed using the proposed localization method based on the fusion of location cues from visual odometry and marker detection. The *orbslam2* trajectory is computed using the publicly available implementation[2] of ORB-SLAM2. The *odometry* trajectory is computed using visual odometry used in the fusion, i.e., it is the *fusion* trajectory without global corrections. The last two columns in Table 4.1 show the number of ground markers detected for each sequence and the distance traveled estimated with visual odometry.

For sequences DM01-DM09, *orbslam2* showed superior performance by having the lowest error on 7 out of 9 sequences. Without any initial information about the environment, ORB-SLAM2 built the map of the environment and managed to produce a very accurate trajectory on all sequences. The dominance of *orbslam2* in sequences DM01-DM09 is to be expected since the images satisfy the assumption of a static environment with non-reflective surfaces. On the other hand, the accurate *orbslam2* trajectory is computed at a much lower frame rate of the images due to the frequent track losses in real-time runs. The *odometry* and *fusion* trajectories were able to produce pose estimates with a frequency slightly higher than 30Hz. For most sequences, the *fusion* trajectory had lower error than the *odometry*, demonstrating the improved accuracy of pose estimation using the marker detection algorithm. The error differences between *fusion* and *odometry* are small, and on DM03 the *odometry* even achieved the best performance, but these results are due to the short recordings in which the traveled distance is almost always below 30 meters. Although the frequent marker detections that corrected the pose every 6-8 meters could improve the estimate of the pose, the *fusion* trajectory had a lower error on 6 out of 9

[2] Code used for evaluation is available at https://github.com/raulmur/ORB_SLAM2

Table 4.1: The offline trajectory results for the Dortmund dataset. The first three rows show the absolute trajectory error in meters for each sequence, the fourth row shows the total distance traveled during the recording, and the last row contains the number of detected ground markers (note that 2 markers are always detected at the start and end of sequence).

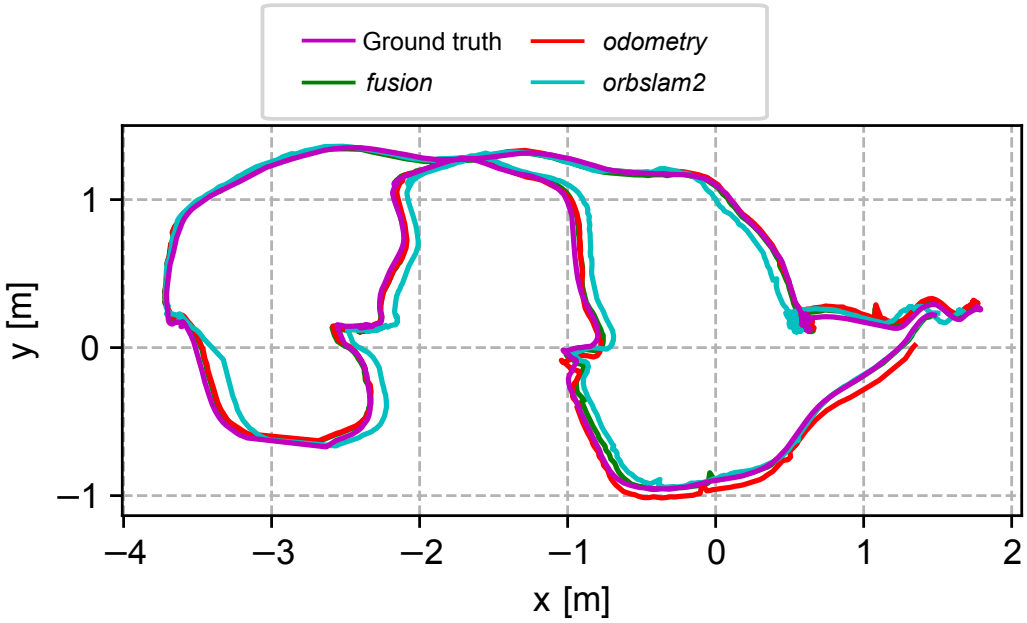
ATE [m]	*fusion*	*odometry*	*orbslam2*	distance	markers
DM01	**0.044**	0.058	0.120	24.0	4
DM02	0.098	0.122	**0.057**	32.4	5
DM03	0.107	**0.053**	0.057	20.4	3
DM04	0.098	0.115	**0.029**	22.5	5
DM05	0.104	0.100	**0.022**	25.8	5
DM06	0.072	0.141	**0.038**	20.6	5
DM07	0.051	0.135	**0.032**	20.3	8
DM08	0.066	0.070	**0.025**	25.0	3
DM09	0.091	0.080	**0.020**	18.0	3
DM12345	**0.185**	0.678	0.550	125.1	22

sequences. Moreover, in the worst case, the *fusion* trajectory had an error 8.2 cm higher than the *orbslam2*. This shows that the proposed fusion method is able to perform close to the state-of-the-art SLAM under static environment conditions in the Dortmund dataset.

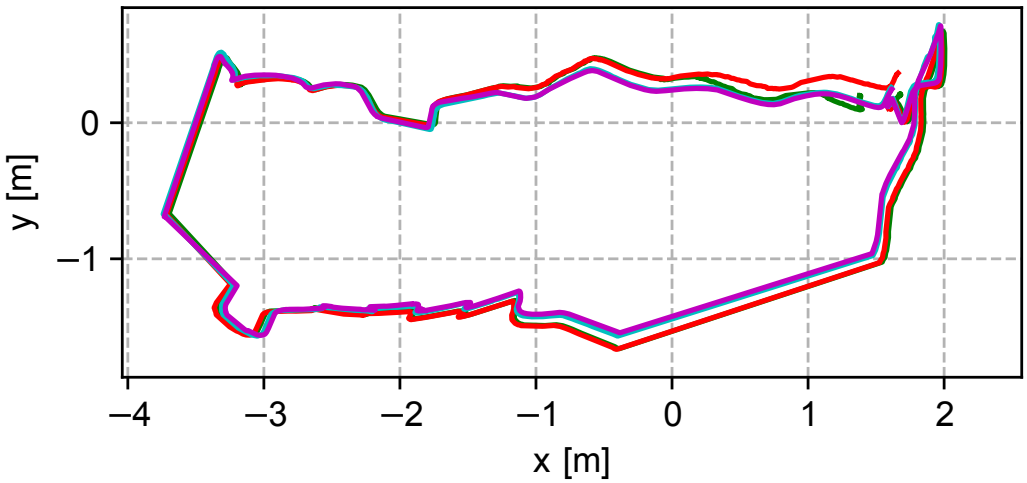
The qualitative evaluation is performed using plots of the trajectories for sequences DM01 and DM09, Fig. 4.6. The lowest error of the *fusion* trajectory on DM01 can be seen in Fig. 4.6a, as it matches almost perfectly with the ground truth trajectory. The *odometry* trajectory has a slightly different shape and the difference between it and the *fusion* trajectory is most pronounced at the bottom part of the trajectory. The misalignment of the *orbslam2* with ground truth comes from the drift in the z-axis, which cannot be seen from the top-down perspective. Nevertheless, the *orbslam2* managed to be closer to the ground truth than other approaches in the sequence DM09, Fig. 4.6b. The straight line in Fig. 4.6b is due to the lack of ground truth in this trajectory segment.

Online trajectory evaluation The trajectories that are corrected during localization have two versions: *offline* and *online*. The offline trajectory is the one computed at the end of the experiment with optimization over all collected measurements. This version of the trajectory is usually evaluated in the popular SLAM benchmarks [6], [103]. On the other hand, for the warehouse worker localization use case, the online trajectory is of more interest than its offline version. The online trajectory is a set of poses created using only the measurements collected up to that point. It does not use future information in the optimization process, which is a logical limitation for the online localization process. The difference between the online and offline trajectories is best seen in the moments before and after the correction, i.e., loop closures, where the offline trajectory has a smooth transition and the online trajectory has a discontinuity.

The results for the online trajectories are shown in Table 4.2. As expected, the online trajectories perform worse than their offline versions. On sequences DM01-DM09, the



(a) Sequence DM01



(b) Sequence DM09

Figure 4.6: The top-down view of trajectories for two sequences from the Dortmund dataset. Descrete jumps are caused by lack of ground truth data.

online version of the *fusion* trajectory kept the absolute trajectory error below 0.17 meters, while the online *orbslam2* trajectory had a larger error on sequence DM01. The error of online *orbslam2* on DM01 comes from the deformation due to the erroneous orientation estimation, which was corrected afterwards and therefore is not present in the offline *orbslam2* trajectory. The differences between the online and offline *orbslam2* trajectories and ground truth are shown in Fig. 4.7.

NON-STATIC ENVIRONMENT SCENARIO. The environments of the automated warehouses are not static because the racks are moved during the process in the warehouse. This is a special localization example, since the static environment is the basic assumption of the SLAM approaches. In the Dortmund dataset, the racks were simulated with the plastic boxes and it was not possible to rearrange the layout during the sequence recording.

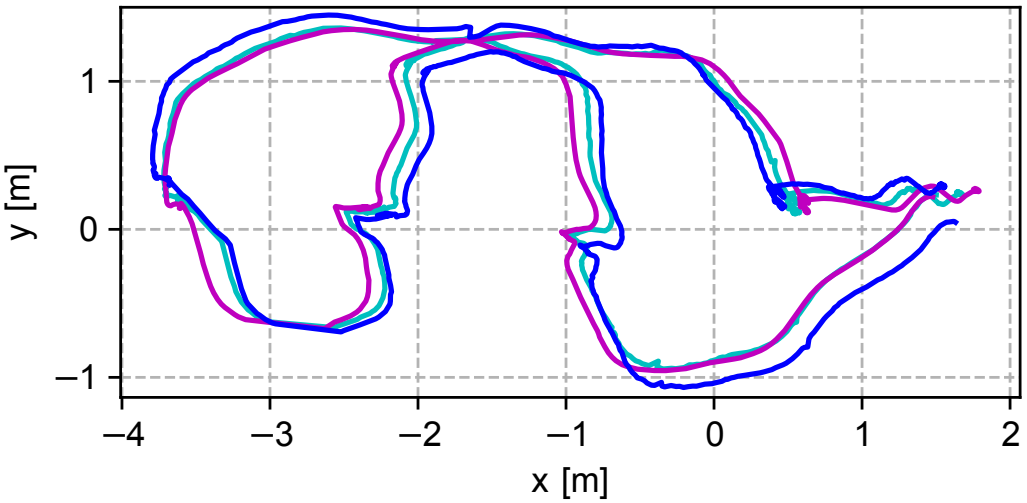


Figure 4.7: Comparison of the online (blue) and the offline (cyan) *orbslam2* trajectory with ground truth (purple) for sequence DM01.

Table 4.2: The online trajectory results for the Dortmund dataset. The table shows the absolute trajectory error in meters for each sequence.

ATE [m]	*fusion*	*orbslam2*
DM01	0.073	0.244
DM02	0.109	0.108
DM03	0.144	0.101
DM04	0.095	0.047
DM05	0.158	0.031
DM06	0.104	0.049
DM07	0.092	0.026
DM08	0.069	0.036
DM09	0.167	0.030
DM12345	0.268	0.592

Nevertheless, the scenario of a non-static environment is simulated by merging several sequences with different rack positions. Sequences DM01-DM05 were merged, and since they all start and end at the same location, it was possible to chain them together without losing a location estimate. Special care was taken to ensure that in all cases all approaches could track features between the images at the end of the former and the beginning of the new sequence, which is critical for continuous location estimation.

In Table 4.1, the last row shows the results of the merged sequences. The proposed approach achieved the best result because the absolute trajectory error was 36.5 cm lower than that of ORB-SLAM2. Also, in Fig. 4.8 we show the absolute error of the proposed solution and ORB-SLAM2. From this we can see that the proposed method was closer to ground truth most of the time. Due to the stochastic nature of ORB-SLAM2, different results are obtained in several runs, so the errors of 4 different runs are shown. In one of the runs, at about 25 s, ORB-SLAM2 lost track of the features and this period is marked with

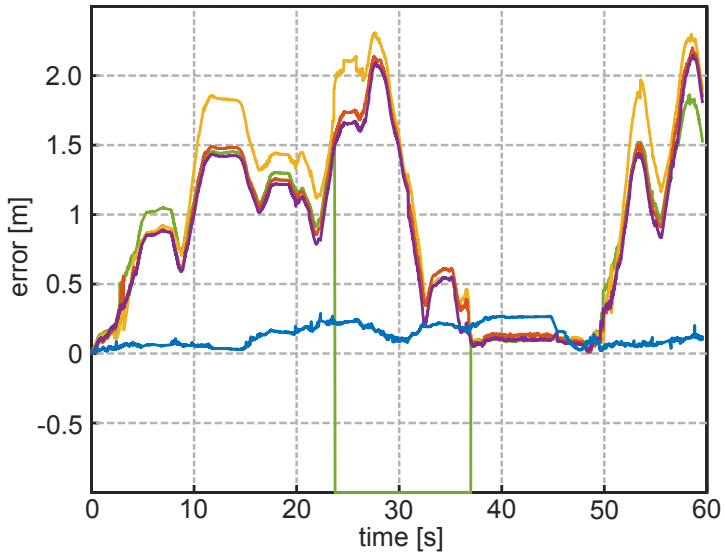


Figure 4.8: The absolute position error in time of the proposed solution (blue) and multiple runs of ORB-SLAM2 on the sequence DM12345. The value −1 signifies that no pose estimates are produced by ORB-SLAM2 due to losing tracks of features.

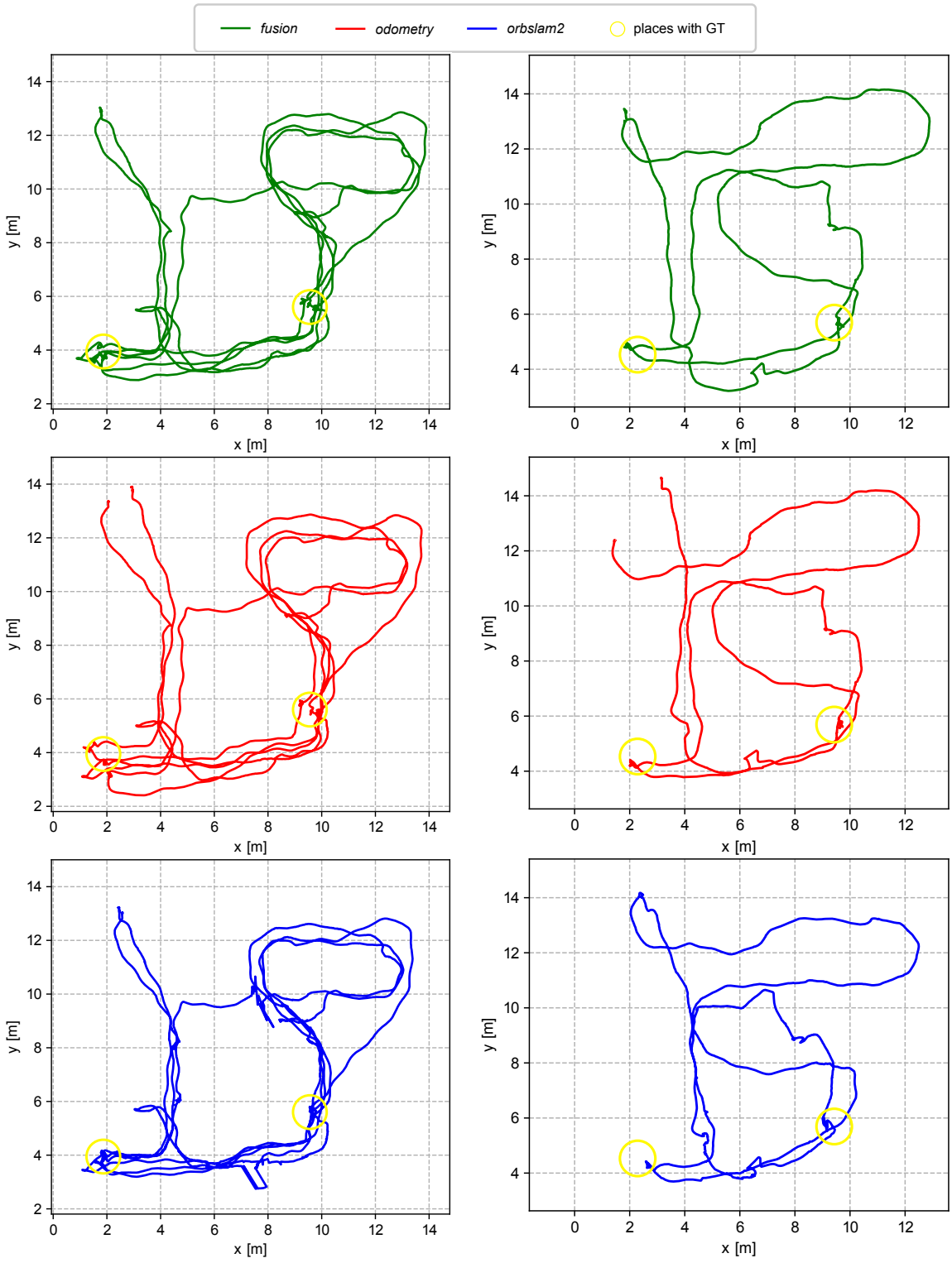
the value −1. The last row of Table 4.2 shows that the online *fusion* trajectory had a smaller error than the online *orbslam2* also in the non-static scenario.

Augsburg dataset

The Augsburg dataset was recorded in a test arena that closely resembles a real automated warehouse. The localization area, enclosed with the security fence, is $12 \times 13\ \mathrm{m}^2$ and filled with metal racks. For this dataset, the collection of ground truth data was a difficult task for the reasons mentioned in Section 2.3.2.2, which resulted in ground truth data for only two small areas in the test arena. Ground truth accuracy in these areas is estimated to be less than 20 centimeters.

The results of the dataset are divided into three scenarios. The first is the *standard operating conditions* scenario, which contains 4 sequences in which the human walks in the warehouse and performs typical tasks. The second scenario is the *kidnapped human* scenario where cameras were briefly covered to simulate a situation where the sensors' field of view is obstructed and localization fails. Finally, the third scenario is the case of a *non-static environment* where the racks were redistributed during the sequence.

STANDARD OPERATING CONDITIONS SCENARIO. The evaluation of this scenario is performed using the three trajectories (*fusion*, *odometry*, and *orbslam2*) as in the Dortmund dataset. An example of the computed trajectories for two sequences AG01 and AG03 can be found in Fig. 4.9. The yellow circles in the images show the areas where the ground truth pose is available. The limited availability of ground truth makes it difficult to compare the performance of the methods based on quantitative analysis alone. On the other hand, quantitative analysis of two ground truth sections combined with a visual inspection of the trajectories makes a good indicator of method performance.



(a) AG01 (b) AG03

Figure 4.9: Augsburg dataset - trajectory examples. The ground truth was not available along the whole trajectory, but only at the two sections marked with yellow circles. All sequences begin and end at the same position.

Table 4.3: The results for the Augsburg dataset. The second, third and fourth columns show the absolute trajectory error in meters, the fifth column shows the total distance traveled for each of the recordings in meters, and the last column is the number of detected markers for the sequence.

ATE [m]	*fusion*	*odometry*	*orbslam2*	distance	markers
AG01	0.328	0.548	**0.128**	170.6	7
AG02	**0.191**	0.327	0.514	140.9	4
AG03	**0.303**	**0.303**	0.661	83.9	3
AG04	0.719	0.661	**0.532**	117.0	5

Table 4.3 shows a mixed dominance of ORB-SLAM2 and the proposed solution. For example, on the sequence AG01, Fig. 4.9a, the *orbslam2* trajectory has the lowest error, while the worst performance is obtained from the *odometry* trajectory, which drifted away, as can be seen in the upper left corner of the red trajectory in Fig. 4.9a. Although the ratio between the distance traveled and the markers detected is high on sequence AG02, the *fusion* trajectory achieved the best performance. The reason for this is the detection of a ground marker near the evaluation zone. A trajectory with marker detections close to the evaluation zone has a lower impact of the odometry error on the result. Furthermore, not only the number of detected markers is important, but also the accuracy of the marker pose estimate. For example, if the marker detection algorithm provides a pose estimate with a correct position but a large orientation error, the fusion of such a pose estimate will result in a translation error in the subsequent pose estimates with odometry. Another interesting example is the sequence AG03, shown in Fig. 4.9b, where *odometry* and *fusion* have the same score, which is lower than that of *orbslam2*. Although the odometry was fairly accurate up to the ground truth parts, it still drifted at the end, as can be seen in the upper left corner of the image. Finally, the results for AG04 indicate similar relative performance of the algorithms as in the AG01 sequence.

The results of the evaluation of the online trajectories are shown in Table 4.4. The online *fusion* trajectories showed higher absolute trajectory error than the offline *fusion* trajectories on all but one sequence. On sequence AG03, both the online and offline *fusion* trajectories have the same error. The same error value results from the ground truth locations being traversed entirely in a single interval between the detection of two ground markers. In this case, the interval started with the marker detected at the beginning of AG03, so the first parts of the trajectories overlapped Fig. 4.10. On the other hand, the online *orbslam2* trajectories had a lower error, which is rather unusual. The unusual results of the quantitative analysis should be attributed to the fact that ground truth is sparsely available and its accuracy is estimated to be less than 20 centimeters.

KIDNAPPED HUMAN SCENARIO. The *kidnapped robot* problem in mobile robotics is a situation where the robot is taken during the localization process and placed to an arbitrary location. Based on the measurements after relocation, the robot has to determine its new location. Similarly, when localizing humans in a warehouse, there may be the problem of the kidnapped human. Such a problem may occur when the location cues are obstructed,

Table 4.4: The online trajectory results for the Augsburg dataset. The rows show the absolute trajectory error in meters for each sequence.

ATE [m]	*fusion*	*orbslam2*
AG01	0.506	**0.075**
AG02	0.277	**0.268**
AG03	0.303	**0.112**
AG04	0.740	**0.241**

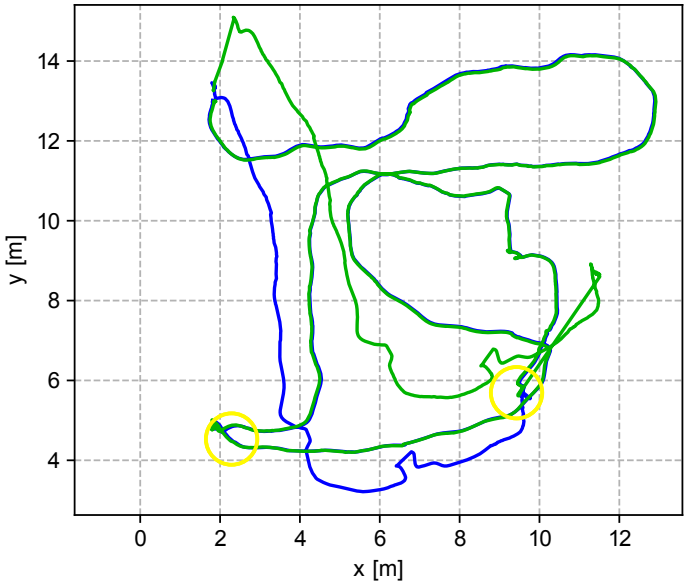


Figure 4.10: Comparison of the online (green) and the offline (blue) *fusion* trajectory on sequence AG03. The first marker detection, detected by discrete jumps in online trajectory, appeared after visiting the areas with ground truth marked with yellow circles.

i.e., the input images from the camera on the Safety Vest become unusable.

In the experiment, the kidnapped robot problem is simulated by covering the cameras with hands while walking through the test area of the warehouse. The lack of visual input disables pose estimation with visual odometry and marker detection. Once the cameras are uncovered, both algorithms continue to estimate the pose, but under the assumption that the human did not move during the blackout phase. This scenario is illustrated in Fig. 4.11, which shows the *odometry* and *fusion* trajectories for four moments in the sequence. The *fusion* trajectory is colored blue, while the *odometry* trajectory is green. The racks in the test arena are shown with red obstacles. At the beginning, before the cameras are covered, Fig. 4.11a, both trajectories are similar and the visible differences come from the marker detection algorithm corrections (indicated by a red arrow). The moment when the cameras are covered is shown in Fig. 4.11b, and the pose is marked with the dashed orange circle. During the blackout phase, the human moved from the lower left side to the lower right side of the lower three racks, the orange arrow in Fig. 4.11c. After the cameras were uncovered, both the visual odometry and the marked detection algorithm continued their normal operation with the old pose estimate. Figure 4.11d shows a correction of the blue trajectory when a ground marker is detected. Although the initial detection does not fully correct the

pose, successive corrections make the position of the proposed method more accurate and the trajectory ends near the starting point, as shown in Fig. 4.11e. We assume that this is the true pose since all sequences start and end at the same point.

NON-STATIC ENVIRONMENT SCENARIO. The safety system based on relative ranging was still under development at the time this dataset was recorded, and it was not possible to safely perform the localization experiment in the test facility as the robots moved the racks around. Therefore, as in the Dortmund dataset, the redistribution of racks during localization is simulated by stitching together sequences with different rack arrangements. Again, all sequences start and end at the same location and for all methods evaluated, the pose was successfully tracked across the merged parts of the sequence. The sequences with different rack distributions were recorded on two separate visits to the test facility, and only the recordings from the second visit contained the ground truth based on AprilTags. Therefore, it was not possible to reliably test accuracy as in the scenario with standard operating conditions. However, the performance evaluation in this case is done by tracking the number of times the localization method failed to provide a pose estimate. Namely, when the localization fails, both the proposed approach and ORB-SLAM2 stop sending pose estimates. The proposed approach waits for the next successful feature tracking from visual odometry, while ORB-SLAM2 has to relocalize in the created map of the environment. Therefore, missing pose estimates can serve as an indicator of localization reliability. In these experiments, this metric is used to evaluate the performance of the methods.

All merged sequences start with the same sequence recorded during an earlier visit to the test facility, and then have one of the sequences used in the evaluation of the standard operating conditions scenario. The ground marker map remained unchanged during the two visits, while the position of the racks was changed. The hypothesis is that the changes in the environment adversely affect the ORB-SLAM2 trajectory estimate. Unfortunately, only on two of four sequences, AG02 and AG03, both algorithms managed to continue localization after the transition between sequences. The localization loss results are shown in Fig. 4.12. Due to the stochastic nature of ORB-SLAM2, it is run 5 times to determine the average performance. The figure shows that the proposed approach has no localization losses. On the other hand, the mean value ORB-SLAM2 losses for 5 runs was 16.1% and 11.9% for the merged AG02 and AG03, respectively. For the record, on the original testing facility sequences only ORB-SLAM2 has localization losses, for example 11.4% on AG02.

The gaps in ORB-SLAM2 change for different runs, but there are 4 repeating gaps in 5 runs of the AG02 sequence and 2 repeating gaps in 5 runs of the AG03 sequence. The gaps in Fig. 4.12 for the different runs of ORB-SLAM2 are almost congruent, and Table 4.5 shows for one of these runs how much the fusion pose estimate moved while ORB-SLAM2 was unable to localize. The results in Table 4.5 show that the gaps were not due to a situation where the stereo camera was held still in a visually difficult area where ORB-SLAM2 lacked features, and the proposed algorithm showed to be more robust. Thus, we can conclude that the proposed algorithm shows more reliable localization performance in changing environment conditions, that are expected in robotized warehouses.

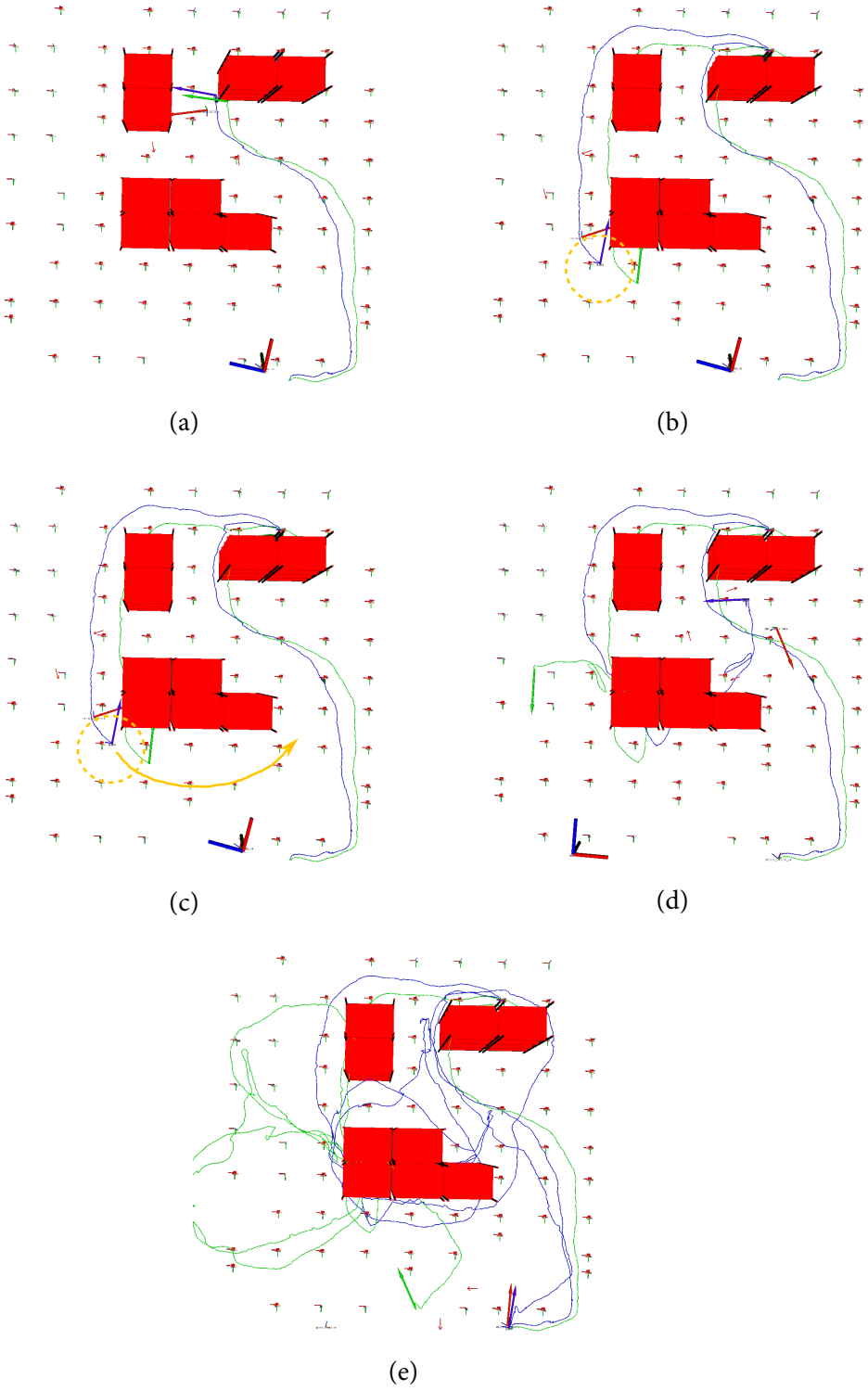
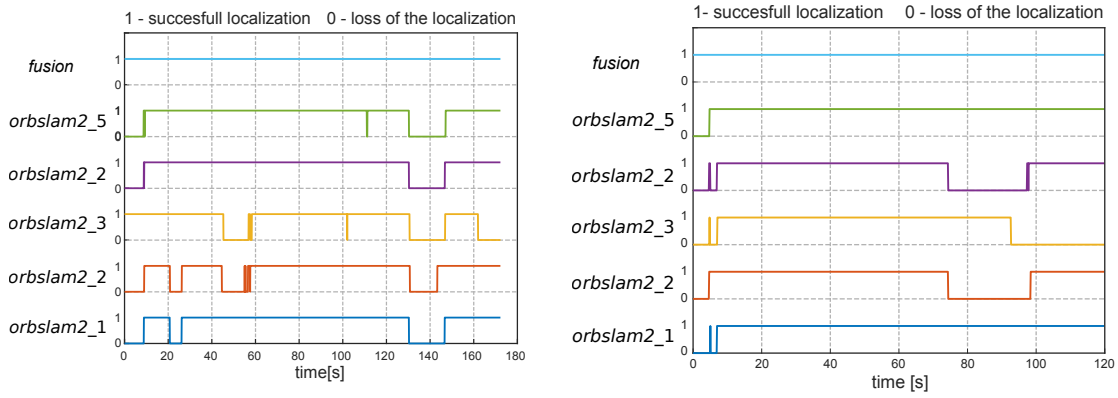


Figure 4.11: The kidnapped-human scenario captured in four different time steps. Red squares represent the racks present in the testing arena. Blue and green trajectories with the corresponding pose arrows represent the *fusion* and the *odometry* estimations respectively. The red arrow represents the last detected pose with the ground stickers detection. With help from the sticker detection, the blue trajectory manages to return to the starting position, whereas the green trajectory drifts away.

Table 4.5: The localization gap intervals and the distances between poses of the *fusion* trajectory at the beginning and the end of each interval.

Sequence	Interval [s]	Distance [m]
AG02	0 - 8.2	1.8
	20.3-25.7	6.2
	44.1-56.9	6.6
	127.3-143.3	5.8
AG03	0 - 7.0	2.3
	74.3-97.4	7.2



(a) AG02 merged

(b) AG03 merged

Figure 4.12: Loss of localization in time for 5 different runs of ORB-SLAM2 and the proposed algorithm on the merged sequences AG02 and AG03.

SUMMARY

In this chapter, a novel method for human localization in automated warehouses is presented. First, the SafeLog project is presented to illustrate the motivation and constraints of the use case, which limited the choice of sensors and consequently shaped the proposed localization method. The approach is based on wearable visual sensors, a downward-looking monocular camera and a horizontally-oriented stereo camera. The warehouse has a set of known pose markers evenly distributed on the ground, and the monocular camera computes the global pose estimate by detecting the markers. In addition, the stereo camera is used to compute the pose estimate with visual odometry, and the proposed method combines both estimates within a graph optimization framework. After presenting the methods for pose estimation and the fusion process, the proposed method is evaluated on self-recorded datasets and compared with the state-of-the-art SLAM ORB-SLAM2. The datasets are evaluated in two scenarios: the standard operating conditions scenario and the non-static environment scenario. In the standard operating conditions scenario, a human walks through the warehouse and performs various movements expected during normal operations. The scenario with a non-static environment additionally simulates the situation in which the position of the racks change during the localization. In addition, a kidnapped human experiment is performed where the visual input was disabled by covering the cameras during the localization process and the proposed method managed to correct its pose with the global

pose corrections once the cameras were uncovered. The dataset results showed that the proposed approach yields a robust and real-time localization with accuracy comparable to ORB-SLAM2 without requiring any modifications to the existing warehouses. Moreover, compared to ORB-SLAM2, the proposed approach is computationally lighter and more robust to changes in the environment that may occur frequently in robotized warehouses as the robots redistribute and carry the racks to the picking locations.

5

Robustness improvement of stereo visual localization using UWB sensors

The challenging problem of localizing people in warehouses based on visual sensors was described in the previous chapter. Scenes with low texture, badly illuminated areas, and blurred images can degrade the quality of the information provided by visual sensors, which in turn affects the accuracy of the estimation. In addition, the quality of the information may not be consistent throughout the estimation process because the lighting conditions throughout the warehouse are not the same, nor are the movements of the human carrying the visual sensors. In addition, the challenging visual conditions during the localization process may result in a loss of location estimate and thus degrade the overall localization performance. In this chapter, modifications to the previously proposed localization method are presented to improve its robustness in scenes with poor visual information. Two changes are introduced: First, a non-visual sensor is added to improve location estimate under visually challenging conditions, and second, the quality of visual information is considered during the optimization process. The changes are tested on the Dortmund, Augsburg and Zagreb datasets, and compared with the originally proposed localization method presented in the previous chapter and with the state-of-the-art SLAM method ORB-SLAM2.

Motivation

In the previous chapter, a method for localization of human workers in automated warehouses was proposed using only visual sensors, a stereo camera and a monocular camera. The conditions in the warehouse can be controlled to some extent, but during the usual work of the human, many reasons can cause poor image quality of the cameras. If the camera's field of view is blocked because a worker is standing too close to a shelf or other low-texture surface, or if images are blurred because the worker is walking too fast in a shaded corridor, this can affect visual location estimate. However, in these visually challenging scenarios, the robustness of localization can be increased by adding non-visual sensors to the existing visual ones. In general, sensors in localization systems are divided into active and passive sensors [110]. Active systems, such as Radio Frequency Identification (RFID), Ultra-Wideband (UWB), and Bluetooth, operate in an anchor-node configuration, where the position of a node is determined based on its communication with the network of known position anchors. In contrast, passive sensors, such as cameras and LIDARs, do not require a network

of anchors as part of the positioning system. In recent years, UWB sensors have been widely used in research for indoor localization because they can provide range measurements with centimeter-level accuracy when line-of-sight is available. In addition, the accuracy of distance measurements is not affected in indoor environments with reflective surfaces, such as metal racks in a warehouse. This popularity led to robust methods for localization in complex indoor environments [111], [112], [113]. In [111] authors proposed localization with UWB nodes based on the particle filter. The property of the particle filter to describe multimodal distributions was used to cover the scenarios where the position ambiguity was caused by multipath effects in UWB ranging without line-of-sight. In the case of a complex scenario with a limited number of UWB nodes, it may be difficult to obtain three line-of-sight measurements. Therefore, in [113] authors presented a method that weights the line-of-sight measurements differently from the non-line-of-sight measurements. In addition, the authors present the fingerprint dataset that provides additional measurements for the trilateration when not enough measurements are available. In [112] authors provide an overview of UWB localization methods with a focus on indoor multi-UAV localization.

The idea of combining UWB sensors with other sensors was presented in [114], [115], and [116]. In [114] authors integrated UWB and an inertial navigation system into a factor graph to reduce indoor positioning errors of unmanned aerial vehicles, while the authors in [115] used filtering techniques to fuse the information from IMU and UWB cues. The fusion of the UWB network and laser rangefinders in [116] is used to localize a robot in a changing environment with very accurate positioning near docking stations. In addition to localization, the UWB sensor network can also be used for collision avoidance in multi-robot environments such as automated warehouses [117]. Similarly, the safety system developed in the SafeLog project, where all robots and human workers are equipped with UWB nodes, ensures the safety of human workers by stopping robots that come too close to them. In addition, the UWB node on the Safety Vest makes the ranging measurements to the robots nearby, and the warehouse location server knows the poses of all the robots in the warehouse. This information can be used to compute the position of the Safety Vest.

The proposed localization method computes the relative pose estimate with visual odometry, and in the previous chapter the graph optimization trusts equally to all relative estimates. However, the odometry estimate progressively degrades with time, and assuming that the average speed of the worker is constant, the relative estimate of a 1-meter trajectory has less uncertainty than the estimate of a 100-meter trajectory. In addition, some parts of the trajectory have more appropriate conditions than the others and, accordingly, the corresponding parts of the trajectory have higher uncertainty in the pose estimate. The proposed localization method in the graph optimization framework would benefit from a model that provides the uncertainty value for each relative pose estimate, i.e., the edge of the optimization graph. The part of the trajectory with higher uncertainty would be more deformed after the optimization process. The odometry uncertainty modeling has been studied in [118], [119], [120]. The uncertainty of the stereo visual odometry pose estimate coming from the locations uncertainty of features used for matching was studied in [118], where authors assumed a Gaussian uncertainty distribution of features' 3D position and tracked the uncertainty using the Kalman filter. The estimated uncertainty of each feature was taken into account when estimating the global pose estimate and the global pose

estimate uncertainty. A similar idea is presented in [120], where authors derived an analytical expression for computing the covariance matrix of a stereo visual odometry. Another error model for visual odometry is presented in [121], where authors derived analytical expressions for a 3D feature location uncertainty in measurements of a RGB-D sensor, and propagated it through the visual odometry solution to get the expected uncertainty bounds around the estimate. In [119] author modeled the odometry error for a mobile robot with synchronous drive using four parameters that relate to systematic and non-systematic errors. In [122] the authors used a convolutional network to learn the uncertainty of visual odometry.

The localization in automated warehouses requires a simple model for the pose uncertainty estimation due to the limited computational power. Furthermore, in cases with insufficent number of features, due to textureless areas or corrupted images, the visual odometry might not compute the pose estimate at all, and will consequently degrade the quality of the pose estimate.

GLOBALLY CORRECT POSE ESTIMATION WITH UWB SENSORS NETWORK

A common way of localization with a UWB sensor network consists of static UWB sensors with known poses (anchors) and a moving UWB sensor (tag) whose position is being determined. The tag communicates with the anchors and the result of the communication is a set of ranging measurements between the tag and the nearby anchors. The pose of the UWB tag is then computed by trilateration with the ranging measurements. In contrast to the standard scenario with the static anchor network, UWB sensors are used in the automated warehouse primarily for safety reasons and all sensors are attached to the moving objects, robots or humans. Nevertheless, the constancy of the position of the anchors over time is not a constraint that trilateration algorithm could take advantage of to estimate the position. Rather, it is a practical solution to know the position of the anchors once the system is calibrated. On the other hand, the automated warehouses have a location server that knows the poses of all the robots in the warehouse, i.e., the poses of the UWB anchors. This information enables localization even when the UWB anchors are moving, and the localization with the UWB nodes is used as an additional reference for the location in the fusion process. The information flow between the Safety Vest, the robots, and the location server is shown in Fig. 5.1. The drawback of the UWB network with moving anchors compared to static anchors is that there is no guarantee of a pose estimate at a certain location. Localization with moving anchors only provides the estimate if it has a sufficient number of ranging measurements. However, the infrequent pose estimate from the UWB sensors incorporated into the fusion framework provides useful information about the globally correct position of the worker and complements the pose estimates from the ground markers. On the other hand, the inclusion of information from the UWB network in the fusion algorithm avoids the scalability problem, since it does not depend only on the UWB anchors.

Estimating the pose of a worker by trilateration requires known coordinates of the anchors (x_k, y_k, z_k) and the ranging measurement r_k between the anchors and the tag worn by the human. For 3-dimensional space, the three anchors are sufficient to constrain the sought pose to 2 points, one of which is the true pose. For n anchors, the human pose

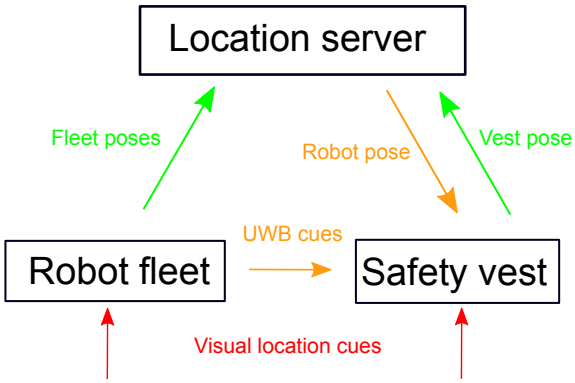


Figure 5.1: Information flow in the proposed localization system. Red arrows show visual cues used for localization of the robots and the human worker. The *Location server* acts as an information broker and the worker localization algorithm can query the location of any robot at any time.

(x_v, y_v, z_v) is found as the intersection of spheres defined by Eq. (5.1).

$$\begin{aligned}(x_1 - x_v)^2 + (y_1 - y_v)^2 + (z_1 - z_v)^2 &= r_1^2 \\ (x_2 - x_v)^2 + (y_2 - y_v)^2 + (z_2 - z_v)^2 &= r_2^2 \\ &\vdots \\ (x_n - x_v)^2 + (y_n - y_v)^2 + (z_n - z_v)^2 &= r_n^2\end{aligned} \tag{5.1}$$

Although the 3 anchors are the minimum number to obtain a finite number of solutions, there may be no solution to such a system of equations for noisy measurements. Therefore, more anchors are used for position estimation. With four anchors, the system of equations in Eq. (5.1) can be converted to a linear system by subtracting the last equation (without loss of generality) from the first three equations, thereby eliminating the nonlinear elements x_v^2, y_v^2, and z_v^2. The remaining system of linear equations in matrix form is given by Eq. (5.2).

$$2\begin{bmatrix} x_1 - x_n & y_1 - y_n & z_1 - z_n \\ x_2 - x_n & y_2 - y_n & z_2 - z_n \\ \vdots & \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n & z_{n-1} - z_n \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} = \begin{bmatrix} r_1^2 - r_n^2 - d_{1n}^2 \\ r_2^2 - r_n^2 - d_{2n}^2 \\ \vdots \\ r_{n-1}^2 - r_n^2 - d_{(n-1)n}^2 \end{bmatrix} \tag{5.2}$$

where $d_{kn}^2 = \left(x_k^2 - x_n^2\right) + \left(y_k^2 - y_n^2\right) + \left(z_k^2 - z_n^2\right)$. The linear system in Eq. (5.2) is solved using the least squares method as in [123]. Although the noisy measurements are filtered by using the average of the distance in a short period of time implemented in a sliding window fashion, some noise remains. The least squares yields the position $\bar{\mathbf{x}} = (\bar{x}_v, \bar{y}_v, \bar{z}_v)$, which minimizes the following error

$$\arg\min_{\bar{\mathbf{x}}} \left(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}\right)^T \left(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}\right) \tag{5.3}$$

where A and b denote the matrix on the left-hand side and the vector on the right-hand side from Eq. (5.2), respectively.

The estimated position of the UWB tag is included in the optimization framework by providing the global corrections similar to the marker detection algorithm corrections. The

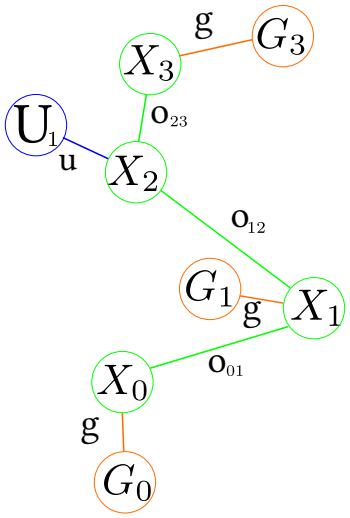


Figure 5.2: The graph constructed with visual sensors and UWB anchors network. The orange nodes represent the global pose estimates computed with the marker detection algorithm. The position of the blue node is computed with ranging UWB measurements through trilateration process, while the orientation is taken from the current orientation estimate. Green nodes represent the Safety Vest estimate and the green edges which connect them are computed with the visual odometry.

difference between the UWB and the ground marker location cue is that the UWB estimates only the position of the tag and not the orientation. Nevertheless, the global position of the UWB tag is valuable information for pose correction. In the graph optimization framework, the position of the nodes representing the pose estimate with the UWB anchor network is estimated by the trilateration process, while the orientation is taken from the current human pose estimate.

VARIABLE ODOMETRY EDGE WEIGHTS IN OPTIMIZATION GRAPH

The optimization graph shown in Fig. 5.2 is used to fuse all location information. The green nodes and variables X_k represent relative localization estimates, e.g., from stereo odometry. The orange nodes G_k and variables g represent global localization estimates coming from the marker detection algorithm, while the blue nodes U_k and edges u come from the trilateration of the UWB ranging measurements. In the previous chapter, the optimization graph contained two different edge types: the odometry edge and the ground marker edge. For simplicity, all edge weights of the same type had the same value, but in the experiments it was found that equal weights of the edges lead to underperforming localization estimate. For example, in the kidnapped human scenario from Section 4.5.2.2, during the covered camera walk, visual odometry was unable to estimate the change in pose and assumed that the pose had not changed. However, in the next optimization cycle, the edge containing the segment with covered cameras had the same significance as the edge containing an estimate computed under good conditions. As a result, localization required more subsequent measurements to converge back to the correct pose estimate.

In this chapter, a change in the edge weights of the optimization graph is introduced,

more specifically variable odometry edge weights. The variable edge weights are computed using a simple uncertainty model that does not affect the runtime performance of the localization algorithm when computational resources are limited. The uncertainty model is based on two simple sources of uncertainty: (i) the magnitude of the distance traveled and (ii) the number of features used for the pose transformation. The first source of uncertainty simply represents the increase in uncertainty as the human worker moves through the environment, while the second source of uncertainty accounts for errors due to low-textured and poorly illuminated scenes where odometry becomes prone to error. Each time new nodes and edges are added to the graph, the weight of the following edge is reset to the maximum value.

Uncertainty due to traveled distance

The first source of uncertainty is based on the fact that the error in odometry estimation increases with the magnitude of motion. Pose error is divided into translation error and rotation error. Figure 5.3 shows a simplified 2-dimensional example of their propagation along the trajectory. In both cases the translation and rotation errors are added at the initial step, and the subsequent steps do not add any additional errors. It can be observed that the initially added translation error in n_0 maintains the same error magnitude in steps n_1 and n_2. The rotation error of n_0 is propagates to steps n_1 and n_2 with the same magnitude, but the additional translation error increases with distance from the first step. Thus, the total error is a superposition of the three error sources. The source of translation error e_{trans} increases proportionally with the magnitude measured using the Euclidean distance. The source of rotation error e_{rot} is proportional to the amount of rotation α, expressed by

$$\alpha_i = 2\cos^{-1}(\Re\{q_{i+1}q_i^{-1}\}) \tag{5.4}$$

where the quaternions q_i and q_{i+1} describe the orientation in two successive steps in the trajectory. The final error source e_{rt} arises from the initial rotation error affecting the subsequent translation error.

The error model for visual odometry is formulated based on the introduced types of error. In the first step, the error is given by Eq. (5.5).

$$e_1 = \overbrace{k_1 d_1}^{e_{trans}} + \overbrace{k_2 \alpha_1 d_1}^{e_{rt}} + \overbrace{k_3 \alpha_1}^{e_{rot}} \tag{5.5}$$

where d_1 is the amount of distance traveled and α_1 is the amount of rotation. The factors k_1, k_2, k_3 map the magnitude of the rotation and translation to the odometry error. The second term $k_2\alpha_1 d_1$ corresponds to the error caused by the initial rotation error, which propagates to a translation error as the human moves. In the next step, the error elements are increased by the new distance d_2 and angle α_2:

$$e_2 = k_1(d_1 + d_2) + k_2\left(\alpha_1(d_1 + d_2) + \alpha_2 d_2\right) + k_3(\alpha_1 + \alpha_2) \tag{5.6}$$

In step n, the magnitude of the error is given by Eq. (5.7).

$$e_n = k_1 \sum_{i=1}^{n} d_i + k_2 \sum_{i=1}^{n} \alpha_i \sum_{j=i}^{n} d_j + k_3 \sum_{i=1}^{n} \alpha_i \tag{5.7}$$

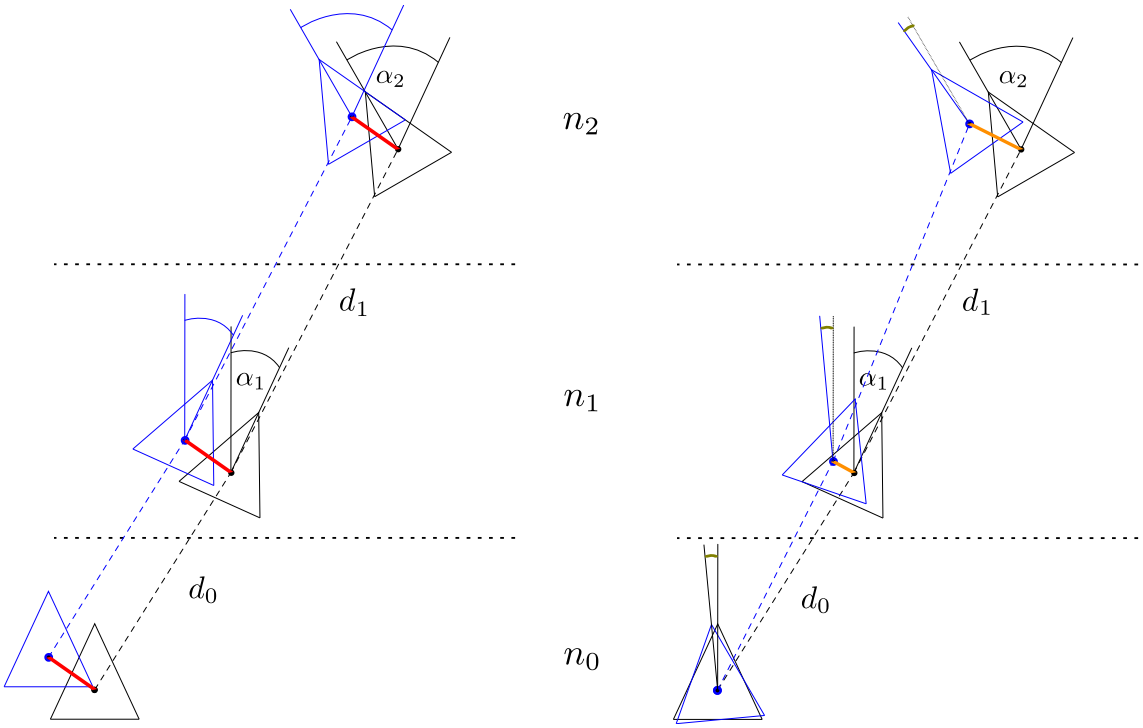


Figure 5.3: A simplified 2D example of error propagation through three consecutive steps. The initially introduced translation error (left) keeps the same magnitude in the following steps. The initial rotation error (right) remains the same rotation error magnitude, but the translation error increases with the distance from the initial step.

Written in recursive form, Eq. (5.7) becomes

$$e_n = k_1 e_n^1 + k_2 e_n^2 + k_3 e_n^3 \tag{5.8}$$

where $e_n^1 = e_{n-1}^1 + d_n$, $e_n^2 = e_{n-1}^2 + d_n e_n^3$, and $e_n^3 = e_{n-1}^3 + \alpha_n$.

With the error e_n computed, we compute the weight of the odometry edge $o_1^{n,n-1}$.

$$o_{n,n-1}^1 = max(O_{\max} - e_n, O_{\min}).$$

The weight is bounded by $O_{\max}$, the weight previously used for all odometry edges, and $O_{\min}$, a small positive value. The error transformation function is shown in Fig. 5.4.

Uncertainty due to lack of features

The second source of uncertainty comes from poorly illuminated or low-textured scenes that are common in warehouses. To account for this effect, the number of features in frames is tracked as human moves through the environment. The model for this source of uncertainty is given in Eq. (5.10).

$$\phi = \frac{sum_of_features}{sum_of_frames} \tag{5.9}$$

$$o_{n,n-1}^2 = \min\left(O_{\max}, \max\left(k_f \phi - f_{\min}, O_{\min}\right)\right) \tag{5.10}$$

where ϕ is the average number of features per frame between two pose nodes and $O_{\max}$ and $O_{\min}$ are the limits of the odometry edge value. The function in Eq. (5.10) is shown in Fig. 5.5.

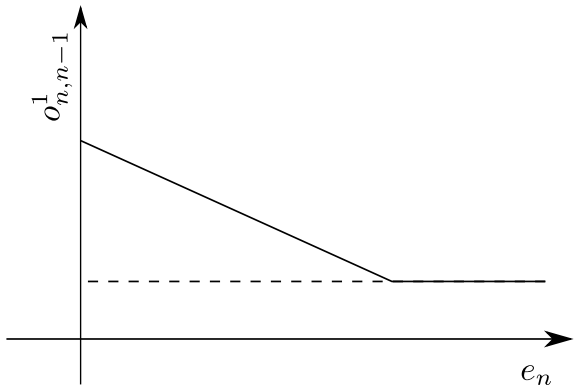


Figure 5.4: A plot of the function which maps the estimated error e_n magnitude into the edge weight $o^1_{n,n-1}$. The initial weight $O_{\max}$ linearly decreases until it reaches the minimum weight value $O_{\min}$.

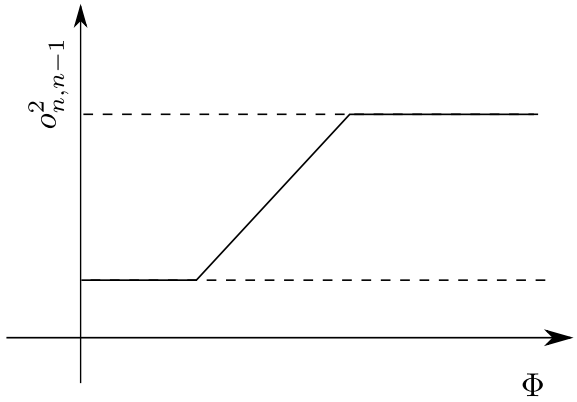


Figure 5.5: A plot of the function which maps the average number of features per frame in the edge Φ to the edge weight $o^2_{n,n-1}$. The weight linearly increases with the number of features per frame between upper and lower limits $O_{\max}$ and $O_{\min}$.

It is assumed that above a certain number of average features per image, the estimate has no lower uncertainty and the function is bounded by the upper bound $O_{\max}$. Similarly, below a certain number of average features per image, the uncertainty of the estimate does not increase and is weighted by $O_{\min}$.

The final edge weighting, which takes into account the total uncertainty, is expressed as the geometric mean of two uncertainty components. The geometric mean is used because of its property of yielding a low edge weight when one of the uncertainty components has a high value.

$$o_{n,n-1} = \sqrt{o^1_{n,n-1} o^2_{n,n-1}} \tag{5.11}$$

The final weight $o_{n,n-1}$ is in the range bounded by $O_{\min}$ and $O_{\max}$.

EVALUATION

The modified localization method, which is aware of odometry uncertainty and can use UWB location cues, was evaluated on three datasets. The Dortmund and Augsburg datasets were already used to evaluate the original localization method in Chapter 4. Unfortunately,

Table 5.1: The values of parameters used for evaluation.

Parameter	Dortmund & Augsburg	Zagreb
Ground marker edge weight	5	5
UWB edge weight	-	1
$O_{\max}$	100	100
$O_{\min}$	1	1
k_f	1.3	1.8
$f_{\min}$	50	50
k_1	0.01	0.01
k_2	0.05	0.05
k_3	0.5	0.5

these two datasets do not have UWB ranging measurements to test localization with UWB location cues. Nonetheless, the evaluation of localization with the UWB anchor network is performed using the dataset previously presented in Section 2.3.2.3, called the Zagreb dataset. In the Zagreb dataset, the poses of the UWB anchors are measured before the experiment and during the experiment the UWB anchors remain static. However, such a setup does not fully represent localization in warehouse environments where the UWB network is dynamic. To account for this, an additional experiment is conducted with the dynamic UWB anchors in the faculty laboratory. Table 5.1 shows the parameter values of the proposed fusion localization approach. The parameters are heuristically determined and the analysis of their appropriate values needs further investigation. In the remainder of the evaluation, the trajectory computed using the originally proposed localization method that uses only visual location cues is denoted as *F-S*, while *F-SU* and *F-SU*$_r$ denote the trajectory that uses all location cues, and the trajectory that uses all location cues and, additionally, has different odometry edge weights in the optimization graph. Lastly, the *F-S*$_r$ trajectory in the Dortmund and Augsburg datasets uses only visual location cues and odometry edge reweighting.

Dortmund and Augsburg datasets

The evaluation of the original localization method is performed on the Dortmund and Augsburg datasets, which provide visual local and global location cues. Due to the lack of a UWB anchor network, it was not possible to evaluate the localization with the fusion of visual and UWB location cues. However, it was possible to evaluate the modification of odometry edge reweighting. Tables 5.2 and 5.3 show the results for the Augsburg and Dortmund datasets, respectively. Table 5.2 shows that the *F-S*$_r$ trajectory performs better on all sequences, indicating that reweighting the edges with the proposed uncertainty model can provide more accurate trajectories. Although the ORB-SLAM2 approach still performed better at 2 sequences, the proposed improvement brought the error of the trajectory closer to that of ORB-SLAM2. Table 5.3 shows that the *F-S*$_r$ trajectory had similar performance to the *F-S* trajectory, mainly due to the fact that in this dataset the sequences were shorter and

Table 5.2: The results for the *Augsburg* dataset. All values are expressed in meters.

ATE	F-S	F-S$_r$	ORB-SLAM2	distance
AG01	0.328	0.270	**0.128**	170.6
AG02	0.191	**0.177**	0.514	140.9
AG03	0.303	**0.130**	0.661	83.9
AG04	0.719	0.712	**0.532**	117.0

Table 5.3: The results for the *Dortmund* dataset. All values are expressed in meters.

ATE	F-S	F-S$_r$	ORB-SLAM2	distance
DM01	**0.044**	0.049	0.120	24.0
DM02	0.098	0.104	**0.057**	32.4
DM03	0.107	0.065	**0.057**	20.4
DM04	0.098	0.084	**0.029**	22.5
DM05	0.104	0.125	**0.022**	25.8
DM06	0.072	0.078	**0.038**	20.6
DM07	0.051	0.050	**0.032**	20.3
DM08	0.066	0.073	**0.025**	25.0
DM09	0.091	0.100	**0.020**	18.0
DM01-DM05	0.185	**0.175**	0.550	125.1

had proper visual conditions, so larger errors could not manifest.

Zagreb dataset

The results of the evaluation on the Zagreb dataset are shown in Table 5.4. Each of the 11 sequences exhibited some type of dominant motion, as listed in Table 2.1 in Section 2.3.2.3. The ORB-SLAM2 was able to compute the trajectories with the least error on 4 of the 11 sequences. However, Table 2.1 shows that the dominant motion in these sequences was relatively simple, while the proposed method proved to be more robust for more challenging sequences. The worst performance of ORB-SLAM2 was on sequences FL04, FL05, and FL09. In these sequences, fast lateral motion and occluded cameras hindered feature tracking, which degraded the quality of localization. In addition, UWB-enhanced trajectory optimization showed better performance than the originally proposed method that used only ground markers and visual odometry. The largest differences were in sequence FL05, where the loss of stereo images occurred and in FL06, where a human spent most of the time walking in the area with UWB location cues. Reweighting the odometry edges showed further improvement in performance by having the lowest error for most of the sequences and in other sequences the difference to the best result was in the centimeter range. However, in the FL10 sequence, the *F-S* trajectory was significantly closer to the ground truth than the *F-SU* and *F-SU$_r$* trajectories due to the erroneous UWB position

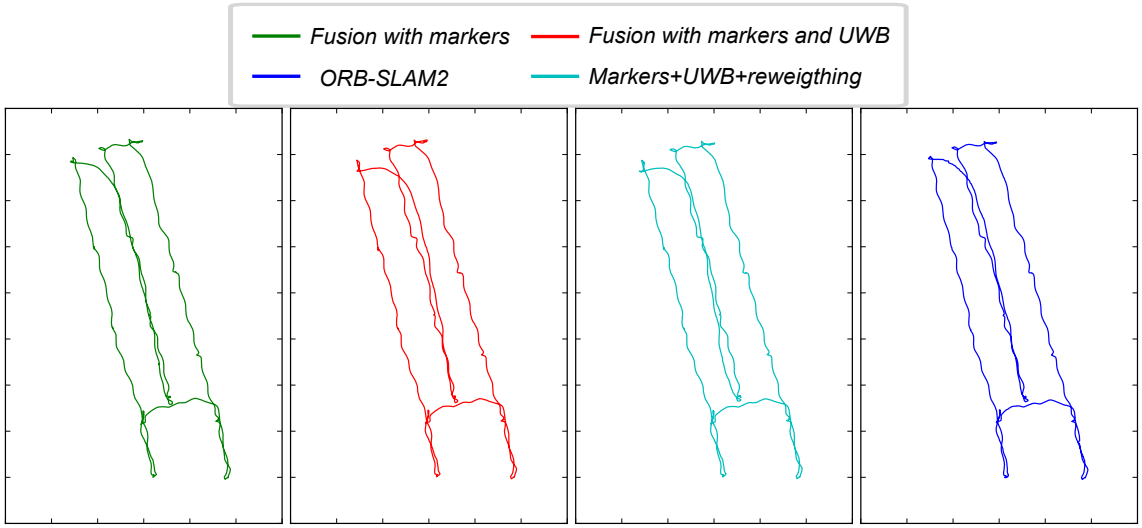
Table 5.4: The results for the *Zagreb* dataset. All values are expressed in meters.

ATE	F-S	F-SU	F-SU$_r$	ORB-SLAM2	distance
FL01	0.088	0.079	0.073	**0.052**	72.1
FL02	0.096	0.055	**0.048**	0.123	62.0
FL03	0.044	**0.036**	0.039	0.049	50.6
FL04	0.174	0.156	**0.109**	2.465	61.9
FL05	4.531	0.263	**0.193**	1.651	76.0
FL06	3.392	0.250	0.264	**0.065**	49.9
FL07	0.151	0.170	0.168	**0.041**	262.6
FL08	0.387	0.108	**0.096**	0.201	101.2
FL09	0.357	**0.200**	0.204	1.434	85.8
FL10	0.215	0.344	0.365	**0.138**	272.6
FL11	0.308	0.249	**0.188**	0.901	109.6

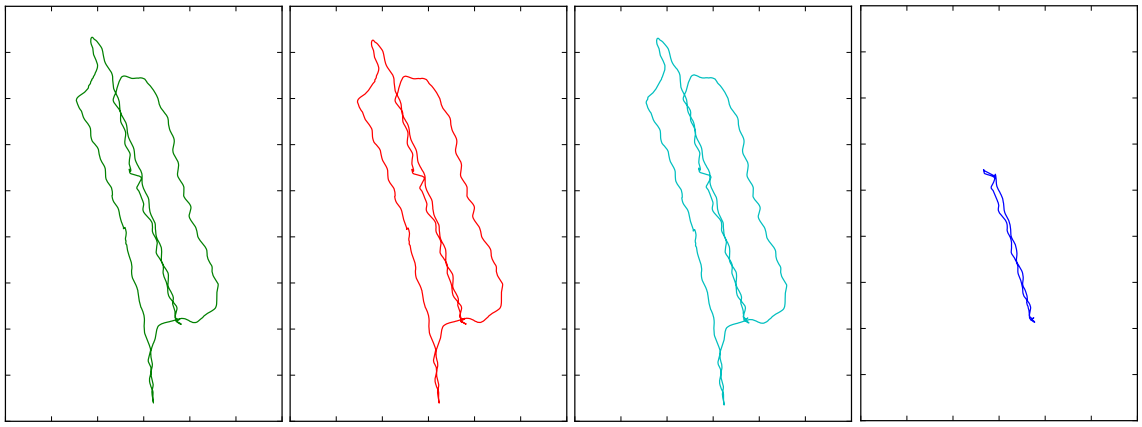
estimate.

The top-down view of the trajectories in Fig. 5.6 shows the qualitative performance of the four algorithms evaluated on the Zagreb dataset. The plots are shown for three sequences: FL01, where ORB-SLAM2 showed the best performance, FL04, where ORB-SLAM2 performed poorly, and FL10, where the original fusion outperformed the other two proposed versions. On sequence F01, Fig. 5.6a, all trajectories are well-formed and the walk through the library corridors, as shown in Fig. 2.13, is clearly visible. The different versions of the proposed method computed very similar trajectories, and none exhibits noticeable deformation. On sequence FL04, all three versions of the proposed method obtained similar results, but for ORB-SLAM2, the lateral walk in FL04 resulted in a loss of feature tracking and map building, leading to a deformed trajectory with high error (see Table 5.4). The results for sequence FL10 show the importance of UWB localization accuracy. Mainly due to the incorrect UWB position estimation, the trajectory of both *F-SU* and *F-SU$_r$* was deformed, while the *F-S* trajectory remained within the corridor boundaries, Fig. 5.6c. On FL10 sequence, the ORB-SLAM2 achieved the best performance, although it was lost in the lower left corner of the library due to poor lighting conditions, resulting in a discontinuity in the lower part of the trajectory.

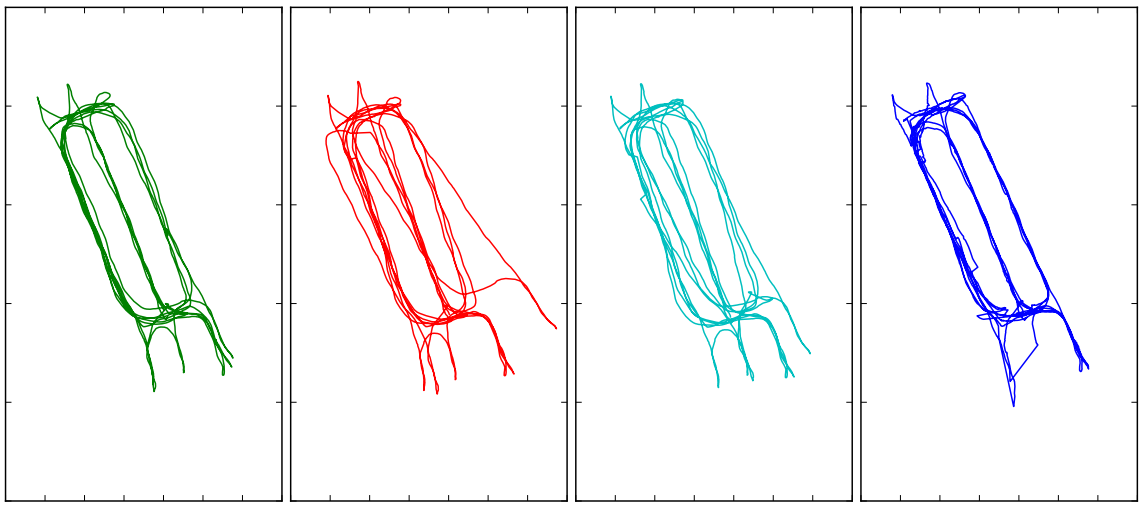
MOVING UWB NODES SCENARIO. The static UWB network in the Zagreb dataset showed the improved performance of localization with the fusion of sensors from different domains compared to localization with only visual sensors. However, the Zagreb dataset only partially simulates the warehouse scenario because UWB nodes are placed on robots in the warehouse and their positions change over time. For this reason, additional experiments are conducted in the faculty laboratory focusing on mimicking the warehouse scenario with moving robots. In these experiments, two moving UWB anchors were introduced, each attached to a moving platform, while another 5 UWB anchors had fixed positions and simulated static robots, Fig. 5.7. These experiments show that localization fusion with UWBs works even if the positions of the UWB anchors change over time. The experiments consist



(a) FL01



(b) FL04



(c) FL10

Figure 5.6: Comparison of four trajectories computed with: fusion presented in the previous chapter, fusion with UWB location cues, fusion with UWB location cues and edge reweighting, and ORB-SLAM2 on sequences FL01, FL04, and FL10.

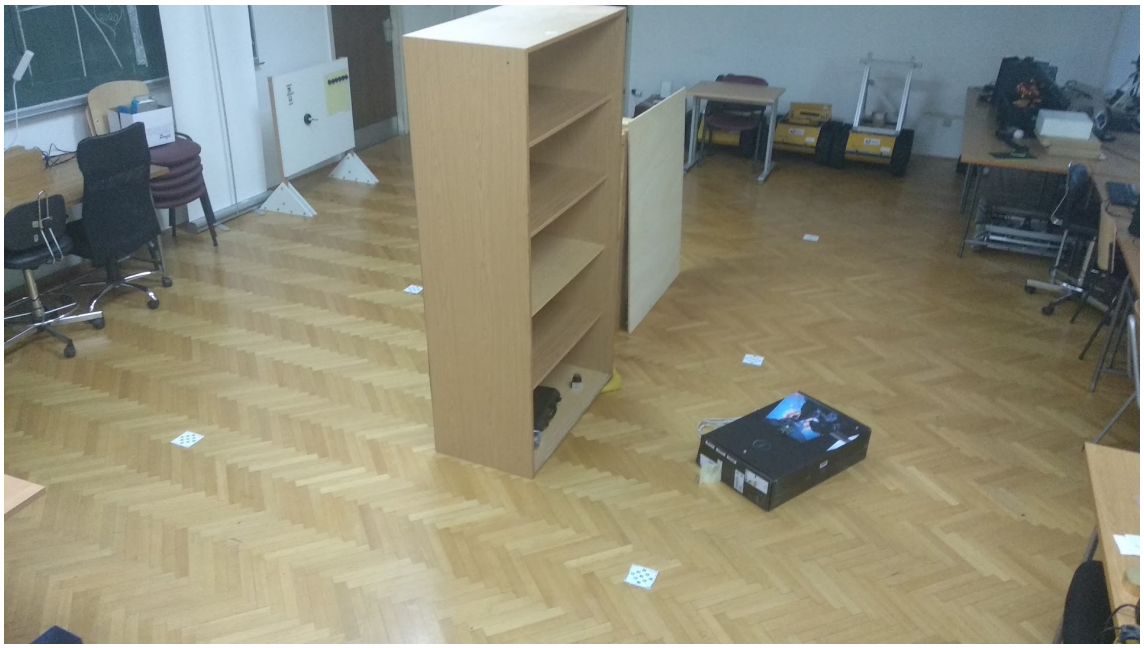


Figure 5.7: A setup for localization experiments with moving UWB anchors. Two moving UWB anchors are placed on the boxes which simulate moving robots.

Table 5.5: Dynamic UWBs localization experiment. All values are expressed in meters.

ATE	F-S	F-SU	F-SU$_r$	F-SU$_{rs}$	ORB-SLAM2	distance
AC01	0.329	0.268	0.225	0.299	**0.076**	56.9
AC02	0.291	0.262	0.192	0.201	**0.148**	62.2

of two sequences, named AC01 and AC02. In sequence AC01, the human localizes in open space, while in sequence AC02 the environment contains an obstacle, which represents a rack. The localization of the robots is simulated with the motion capture system that tracked their poses. In reality, these poses would be provided by the warehouse location server. The motion capture system is also used to record the ground truth pose of the human.

The results of the experiment are shown in Table 5.5. On both sequences, ORB-SLAM2 achieved the lowest error. This result was expected due to the small size of the environment and frequent loop closures. Moreover, the trajectory error of the proposed solution decreased as we added more location cues to the optimization. After the ORB-SLAM2 trajectory, the *F-SU$_r$* trajectory had the second lowest error. The trajectory computed with only 5 static UWB anchors, *F-SU$_{rs}$* , had a higher error than the trajectory that also included the dynamic nodes, *F-SU$_r$*, showing the positive effect of adding dynamic node information.

SUMMARY

This chapter presents modifications to the localization presented in Chapter 4. The modifications are aimed at scenarios where localization based solely on visual sensors may not be accurate enough and improved robustness is needed. The improvement of the robustness of

the localization is due to two reasons: (i) the location information from the non-visual sensors is integrated into the fusion framework, and (ii) the fusion detects segments that might have a low accuracy odometry estimate. The non-visual location information is obtained by trilaterating the range measurements of the UWB sensor network. In the warehouse, both the human workers and the robots are equipped with UWB sensors whose ranging measurements ensure the safety of the workers. The warehouse location server knows the poses of all the robots, and using these poses together with the ranging measurements to localize the human workers requires no additional configuration time or hardware. UWB pose estimation is performed in the warehouse coordinate system, i.e., the global coordinate system, and is equivalent to maker detection localization for the fusion framework, with one exception. Trilateration with UWB range measurements provides only the position estimate, while the orientation is taken from the current pose estimate. Furthermore, a simple model of odometry uncertainty is developed to provide the optimization graph with additional information about the relative transformation reliability. Unlike the previously presented graph that used constant weights for all odometry edges, the modified version uses the developed odometry uncertainty model to compute the weights of the odometry edges. The model focuses on two sources of odometry error. The first comes from the accumulated noise, which is assumed to be proportional to the magnitude of the motion, and the second from the transformation estimates computed with a small number of features.

The Augsburg and Dortmund datasets are used only for the evaluation of edge reweighting with the odometry uncertainty model because they do not contain UWB range measurements. For these datasets, it is shown that considering the odometry uncertainty can lead to better results. On the other hand, both proposed modifications for robustness improvement could be evaluated on the Zagreb dataset, which is equipped with the static UWB sensor network. The evaluation on the Zagreb dataset showed the improved performance of the two proposed components, especially on the difficult sequences where the inference from the visual sensors was either disabled or very difficult due to blur, low texture, or lighting conditions. In addition to the quantitative results on the Zagreb dataset, the sequences of interest are followed by the qualitative results in the form of trajectory plots. On all datasets, the variations of the proposed method showed comparable performance to the state-of-the-art ORB-SLAM2 and superior performance in challenging scenarios that are expected in warehouses. In addition, the experiment in the faculty laboratory included moving UWB anchors and demonstrated the ability of the proposed method to localize in an environment with a dynamic UWB network, such as automated warehouses.

6

Conclusions and outlook

Warehouse automation with robotic systems is being actively developed because of its potential to improve warehouse efficiency and capacity. The active development of robotic systems is accompanied by research in warehouse safety systems, which prevent any unfortunate situation. In automated warehouse systems, such as Swisslog's Carrypick system, a fleet of autonomous mobile robots transports goods stored in heavy racks to the picking stations, where they are taken by human workers. During the process, robots and human workers stay in separate areas, ensuring human safety. The SafeLog project aims to develop a safety concept for similar automated warehouses that improves warehouse efficiency by ensuring the safety of humans while they work in the area with the moving robots. The safety concept is divided into several levels. The safety-critical level shuts down all robots that pose an immediate threat to human safety. The other levels are concerned with avoiding dangerous situations by warning a human worker or redirecting the robots. The sensors on the Safety Vest cover all the functions needed to meet the requirements of the safety concept. The UWB sensors provide accurate and reliable ranging measurements between humans and robots, which is a core part of the safety-critical level. The visual sensors are very informative and are used to make inferences about the worker's environment and location. The contributions presented in this thesis are closely related to the SafeLog safety concept. The methods developed aim to maximize the use of the available environment information due to hardware limitations.

The first contribution of this thesis is a computationally efficient disparity estimation method based on disparity search space reduction. The correspondence problem of disparity estimation methods is usually addressed by searching for similar patches in the stereo images within a predefined search space. The commonly used SGM method estimates disparity by searching for corresponding patches in a predefined range and also enforces the scene consistency constraint. Due to the complexity of the SGM method, it must be run on a GPU or FPGA for real-time applications. Since such hardware requirements are not suitable for the SafeLog use case, the proposed method improves the complexity of the SGM steps by reducing the disparity search space.

While working in the warehouse, a worker carrying a stereo camera will capture a sequence of images and the successive images will mostly capture the same scene. In addition, the scenes in the warehouses contain mostly static objects, as robots with racks and other people are rarely expected in the camera's field of view, and even if they are present, they occupy only a small part of the scene. The complexity of disparity estimation in a

sequence of images containing the same static scene can be mitigated by transforming the disparity estimate from previous frames into the current frame. The transformation of the disparity between frames is based on the method of visual odometry, which determines the displacement of the camera between images from two different time steps. The transformed disparity estimate represents the expected disparity value, and the disparity search space is constrained around this value. The Kalman filter fuses the disparity information from the different steps and keeps track of the uncertainty of the disparity estimate. Based on this uncertainty, the Kalman filter decides which it trusts more in the estimation: the predicted disparity value or the newly estimated disparity. Both the prediction and the new estimate introduce new uncertainties called process and measurement noise, respectively. In addition, the uncertainty of the predicted disparity is used to determine the range of the disparity search space around the expected disparity value. Instead of using heuristically determined values for the process and measurement noise of the Kalman filter, the contribution presents a technique to estimate their value using the training data of the corresponding dataset.

The reduced space disparity estimation method is based on the assumption of a static scene. In the case of a scene where most of the objects are moving, the method will not work because the visual odometry, which predicts the disparity estimate, will give an incorrect transformation estimate. On the other hand, a small number of moving objects will have little or no effect on the disparity prediction. In such cases, the differences between the predicted and newly estimated disparity can be used to detect independently moving objects.

The evaluation of the method is performed using the sequences from the KITTI dataset, since the KITTI Stereo 2015 benchmark does not contain sequential images. The runtime and accuracy of the proposed method are compared to the SGM implementation of OpenCV and LEAStereo, one of the current top ranked disparity methods on KITTI. The methods were compared on 7 sequences and the proposed method showed better accuracy and runtime than the implementation of the original SGM on all but one sequence. The sequence where the proposed method performed poorly contains many moving objects, and further analysis of this sequence showed an increased number of outliers in parts of the sequence with moving objects. On the other hand, LEAStereo was able to achieve much better accuracy on all sequences. However, this method is computationally more complex and the improved accuracy is achieved at the cost of increased runtime. The exact differences in runtime cannot be determined from the presented results because the implementation difficulties led to evaluation on two different processing platforms. Although LEAStereo was evaluated on a much better processing platform, the runtime was higher than the runtime of the proposed method.

Moving object detection is evaluated using the KITTI MOD dataset, which extends the original dataset with the detections of moving vehicles. The evaluation results show that the detection of moving objects in the scene is possible with limited precision and reliability. Since the moving object detection approach is based on detecting differences between the estimated and measured disparity, the bounding box around the moving object is larger than the object itself. Also, the reliability of detection is decreased when small moving objects are present, which sometimes cannot be distinguished from noise.

The second contribution of the thesis presents a method for human worker localization in warehouse environments. An automated warehouse environment is very specific for the

localization problem because the map of the environment is not static. Existing localization solutions are based on a constant appearance of the environment and would have limited success in this application. Moreover, the limited computational resources of the hardware worn on the Safety Vest require a lightweight solution for real-time localization. The proposed solution is based on the fusion of location cues obtained from visual sensors, the stereo camera, and the monocular camera, using the methods of visual odometry and the detection of reliable ground markers. Ground markers already exist in automated warehouses where robots use them to determine their position. The detection of ground markers provides a globally correct pose estimate that cannot be used alone due to the infrequent marker detection. Therefore, ground marker detection is combined with visual odometry information that estimates the relative displacement of the Safety Vest. By merging these two location cues within the graph optimization framework, the method obtains a globally correct pose of the worker at a constant frequency.

Due to the specific requirements for the evaluation of the proposed method, all publicly available datasets were filtered out and the evaluation is performed using self-recorded datasets. The datasets were recorded in warehouse-like environments and simulate a human worker performing the usual activities. In addition, a non-static environment is simulated by stacking together several sequences with different rack layouts, and a kidnapped-human scenario is simulated by covering cameras for a short period of time. The proposed method is compared with the state-of-the-art SLAM method ORB-SLAM2. Both quantitative and qualitative analyses of the results show comparable performance in the standard working scenarios and improved performance of the proposed method over the ORB-SLAM2 in the scenarios with a non-static environment. The kidnapped-human scenario showed that the proposed solution is able to converge back to the correct pose after a period when the visual input was disturbed.

The third contribution of the thesis presents the modification of the proposed worker localization method which improves its robustness under difficult visual conditions. The specific appearance of the warehouse and its lighting conditions are sometimes not favorable for localization with visual sensors. In cases where tracking visual features is not possible due to poor lighting, low-textured scenes, or a limited field of view, estimation of the worker's location is compromised. Therefore, when localizing the worker with graph optimization, it is important to know the quality of the information in the graph. Moreover, in cases where the quality of visual location cues is reduced, greater robustness of the method can be achieved by adding location cues that are not based on visual stimuli. The introduced changes aim at improving the robustness of localization in cases where the quality of visual cues is reduced.

The first change is the addition of a non-visual cue based on UWB sensors to the visual location cues. The UWB sensors are already present on the Safet Vest and autonomous robots, as they are used to determine the human-robot distance in the safety-critical level of the concept. The warehouse management system knows the positions of all the robots, and the UWB sensors measure the distance between the robot and the Safety Vest, so it is possible to determine the worker's position through the trilateration process. The position determined by trilateration represents a new, non-visual location cue that is merged with the other cues in the optimization graph.

The second change is that the optimization graph takes into account the quality of visual odometry. In the optimization, two successive estimates of the human pose are represented by two nodes connected by an edge. The edge represents a transformation between two successive poses estimated with visual odometry, and the weight of the edge represents the certainty of the correct transformation. In the original method, all edges were weighted equally, which led to inadequate results in cases with variable lighting conditions. The weighting of the edge, i.e., the estimate quality, is estimated using a simple error model of visual odometry. The model is based on two sources of odometry error: (i) the amount of motion and (ii) the average number of features per image.

The introduced modifications were evaluated on self-recorded datasets and compared with the originally proposed method and the ORB-SLAM2 method. Since the first two datasets, the Augsburg and Dortmund datasets, used to evaluate the originally proposed method did not contain UWB measurements, they were only used to evaluate the edge reweighting modification. However, an additional dataset was recorded using a UWB sensor network and referred to as the Zagreb dataset. Both modifications were evaluated on the Zagreb dataset. The results showed that the modified version of the proposed localization approach outperforms both the original method and the ORB-SLAM2 method under difficult lighting conditions. Moreover, the performance of both the modified and the original method is comparable to that of ORB-SLAM2 under standard conditions.

The contributions of this thesis deal with two levels of the SafeLog safety concept. Inference about the worker's environment with the stereo camera resulted in a computationally efficient SGM method that outperformed the original SGM on sequences from the KITTI dataset. However, the quality of the disparity still has a place for improvement, as shown by the comparison with the LEAStereo method. In addition, the approach for detecting moving objects needs to be improved, either by reducing noise in the process or by additional post-processing of object detections with one of the tracking methods.

As for worker localization, the proposed method successfully localizes a human worker in a warehouse under different lighting conditions. However, the main challenge of the last two contributions was evaluation due to the unavailability of suitable datasets. To evaluate the proposed method, three datasets were recorded, but only the Dortmund dataset contains almost full coverage with the ground truth pose obtained by the motion capture system since it has a small recording area. During the recording of the sequences in the Augsburg dataset, it was not possible to install the motion capture system that would provide reliable ground truth in the localization area, so ground truth was only available at a few discrete locations. The Zagreb dataset has the ground truth obtained with the motion capture system, but the ground truth area is limited to a portion of the recording area. For this reason, our analysis uses the qualitative results in the form of trajectory images in addition to the quantitative results. Further work should include a new dataset recorded in the arena like in the Augsburg dataset, but with much better ground truth coverage.

BIBLIOGRAPHY

[1] Shimon Y. Nof. *Handbook of Industrial Robotics*. John Wiley; Sons, Inc., USA, 2nd edition, 1999.

[2] IMARC group. *https://www.imarcgroup.com/warehousing-and-storage-market*, April 2022.

[3] IMARC group. *https://www.imarcgroup.com/warehouse-robotics-market*, April 2022.

[4] Swisslog - CarryPick storage and order picking system. *https://www.swisslog.com/en-gb/products-systems-solutions/asrs-automated-storage-retrieval-systems/boxes-cartons-small-parts-items/carrypick-storage-and-picking-system*, April 2022.

[5] Safelog - safe human-robot interaction in logistic applications for highly flexible warehouses. *http://safelog-project.eu/*, April 2022.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Dean Brown. Decentering distortion of lenses. 1966.

[8] Juho Kannala and Sami S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1335–1340, aug 2006.

[9] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[10] Wilhelm Burger. Zhang's camera calibration algorithm: In-depth tutorial and implementation. Technical Report HGB16-05, University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Dept. of Digital Media, Hagenberg, Austria, 05 2016.

[11] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] Berthold K. P. Horn and Michael J. Brooks, editors. *Shape from Shading*. MIT Press, Cambridge, MA, USA, 1989.

[13] Sing Bing Kang, J. A. Webb, C. L. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, page 88, USA, 1995. IEEE Computer Society.

[14] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, apr 2012.

[15] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846. ACM Press, 2006.

[16] Jan Cech and Radim Sara. Efficient sampling of disparity space for fast and accurate matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[17] Payman Moallem, Mohsen Ashourian, Behzad Mirzaeian Dehkordi, and Mohammad Ataei. A novel fast feature based stereo matching algorithm with low invalid matching. *WSEAS Transactions on Computers*, 5:469–477, 03 2006.

[18] Shashi Poddar, Hemraj Sahu, Mohit R Bangale, Vipan Kumar, and Amod Kumar. Feature based dense disparity estimation. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 950–955, 2015.

[19] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001.

[20] Marsha Jo Hannah. Computer matching of areas in stereo images. 1974.

[21] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In Jan-Olof Eklundh, editor, *ECCV (2)*, volume 801, pages 151–158. Springer, 1994.

[22] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pages 648–655, 1998.

[23] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[24] P.F. Felzenszwalb and D.R. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004.

[25] Min Chul Sung, Sang Hwa Lee, and Nam Ik Cho. Stereo matching using multi-directional dynamic programming and edge orientations. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I – 233–I – 236, 2007.

[26] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814 vol. 2, 2005.

[27] Heiko Hirschmüller. Semi-Global Matching Motivation, Developments and Applications. *Proceedings of the 53rd Photogrammetric Week*, (Figure 1):173–184, 2011.

[28] Robert Spangenberg, Tobias Langner, and Sven Adfeldt. Large Scale Semi-Global Matching on the CPU. (Iv):0–6, 2014.

[29] M Agrawal, K Konolige, and L Iocchi. Real-Time Detection of Independent Motion using Stereo. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 2, pages 207–214, 2005.

[30] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016.

[31] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703, 2016.

[32] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015.

[33] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[34] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[35] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[36] H. Laga, L. Jospin, F. Boussaid, and M. Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(04):1738–1764, apr 2022.

[37] Ruan Xiaogang, Yan Wenjing, Huang Jing, Guo Peiyuan, and Guo Wei. Monocular depth estimation based on deep learning:a survey. In *2020 Chinese Automation Congress (CAC)*, pages 2436–2440, 2020.

[38] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1), 2022.

[39] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9, 2017. Publisher Copyright: © The Author(s).

[40] Mohammad Nahangi, Adam Heins, Brenda McCabe, and Angela Schoellig. Automated localization of uavs in gps-denied indoor construction environments using fiducial markers. In Jochen Teizer, editor, *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 88–94, Taipei, Taiwan, July 2018. International Association for Automation and Robotics in Construction (IAARC).

[41] Mohammadreza Yavari. Indoor real-time positioning using ultra-wideband technology. 2015.

[42] Hung Manh La, Ronny Salim Lim, Jianhao Du, Sijian Zhang, Gangfeng Yan, and Weihua Sheng. Development of a small-scale research platform for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1753–1762, 2012.

[43] Christian Landsiedel and Dirk Wollherr. Global localization of 3d point clouds in building outline maps of urban outdoor environments. *International Journal of Intelligent Robotics and Applications*, 1, 12 2017.

[44] Jinwoo Choi, Minyong Choi, Sang Nam, and Wan Chung. Autonomous topological modeling of a home environment and topological localization using a sonar grid map. *Auton. Robots*, 30:351–368, 05 2011.

[45] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008. Semantic Knowledge in Robotics.

[46] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[47] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[48] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[49] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Proceedings of (ICRA) International Conference on Robotics and Automation*, volume 2, pages 1322 – 1328, May 1999.

[50] Hans P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. 1980.

[51] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004.

[52] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, 2011.

[53] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing.

[54] J. Engel, V. Koltun, and D. Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[55] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[56] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[57] Igor Cvišić and Ivan Petrović. Stereo odometry based on careful feature selection and tracking. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2015.

[58] Igor Cvišić, Josip Ćesić, Ivan Marković, and Ivan Petrović. Soft-slam: Computationally efficient stereo visual slam for autonomous uavs. *Journal of field robotics*, 35(4):578–595, 2018.

[59] Bernd Pfrommer and Kostas Daniilidis. Tagslam: Robust SLAM with fiducial markers. *CoRR*, abs/1910.00679, 2019.

[60] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[61] Dorian Galvez-Lopez and Juan D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[62] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision – ECCV 2010*, pages 778–792, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[63] Yao Huang, Fuchun Sun, and Yao Guo. Vlad-based loop closure detection for monocular slam. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 511–516, 2016.

[64] Yi Hou, Hong Zhang, and Shilin Zhou. Convolutional neural network-based image representation for visual loop closure detection. In *2015 IEEE International Conference on Information and Automation*, pages 2238–2245, 2015.

[65] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[66] R. Wang, M. Schwörer, and D. Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[67] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[68] X. Gao, R. Wang, N. Demmel, and D. Cremers. Ldso: Direct sparse odometry with loop closure. In *iros*, October 2018.

[69] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[70] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[71] Carlos Campos, Richard Elvira, Juan J. Gomez, Jose M. M. Montiel, and Juan D. Tardos. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[72] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. Modnet: Moving object detection network with motion and appearance for autonomous driving. *arXiv preprint arXiv:1709.04821*, 2017.

[73] 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*, pages 1280–1286, 2013.

[74] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[75] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[76] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 1989.

[77] Sandino Morales and Reinhard Klette. Kalman-filter based spatio-temporal disparity integration. *Pattern Recognition Letters*, 2013.

[78] Luka Fućek, Ivan Marković, Igor Cvišić, and Ivan Petrović. Dense disparity estimation in ego-motion reduced search space. *IFAC World Congress*, 2017.

[79] Motilal Agrawal, Kurt Konolige, and Luca Iocchi. Real-time detection of independent motion using stereo. In *IEEE Workshops on Application of Computer Vision*, 2005.

[80] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[81] Maziar Loghman and Joohee Kim. Sgm-based dense disparity estimation using adaptive census transform. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 592–597, 2013.

[82] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6640–6649, 2017.

[83] Yeongmin Lee, Min-Gyu Park, Youngbae Hwang, Youngsoo Shin, and Chong-Min Kyung. Memory-efficient parametric semiglobal matching. *IEEE Signal Processing Letters*, 25(2):194–198, 2018.

[84] P. Viola and W.M. Wells. Alignment by maximization of mutual information. In *Proceedings of IEEE International Conference on Computer Vision*, pages 16–23, 1995.

[85] Heiko Hirschmuller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1582–1599, 2009.

[86] Stefan K Gehrig and Clemens Rabe. Real-time semi-global matching on the cpu. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2010.

[87] D. Demirdjian and T. Darrell. Motion estimation from disparity images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 213–218 vol.1, 2001.

[88] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[89] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33:22158–22169, 2020.

[90] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

[91] Robert Spangenberg, Tobias Langner, and Raúl Rojas. Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In *International Conference on Computer Analysis of Images and Patterns*, 2013.

[92] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses, 2008.

[93] Z. M. Bi, Mingchao Luo, Zhonghua Miao, Bing Zhang, W. J. Zhang, and Lihui Wang. Safety assurance mechanisms of collaborative robotic systems in manufacturing. *Robotics and Computer-Integrated Manufacturing*, 67(January 2020), 2021.

[94] Tomislav Petković, Ivan Marković, and Ivan Petrović. Human intention recognition in flexible robotized warehouses based on markov decision processes. In Anibal Ollero, Alberto Sanfeliu, Luis Montano, Nuno Lau, and Carlos Cardeira, editors, *ROBOT 2017: Third Iberian Robotics Conference*, pages 629–640, Cham, 2018. Springer International Publishing.

[95] Farouq Halawa, Husam Dauod, In Gyu Lee, Yinglei Li, Sang Won Yoon, and Sung Hoon Chung. Introduction of a real time location system to enhance the warehouse safety and operational efficiency. *International Journal of Production Economics*, 224:107541, 2020.

[96] Cristina Losada, Manuel Mazo, Sira Palazuelos, Daniel Pizarro, and Marta Marrón. Multi-camera sensor system for 3d segmentation and localization of multiple mobile robots. *Sensors*, 10(4):3261–3279, 2010.

[97] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proceedings Third IEEE International Workshop on Visual Surveillance*, pages 3–10, 2000.

[98] George Chen, John Kua, Stephen Shum, Nikhil Naikal, Matthew Carlberg, and Avideh Zakhor. Indoor localization algorithms for a human-operated backpack system. In *3D Data Processing, Visualization, and Transmission*, volume 3, 2010.

[99] Timothy Liu, Matthew Carlberg, George Chen, Jacky Chen, John Kua, and Avideh Zakhor. Indoor localization and visualization using a human-operated backpack system. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, 2010.

[100] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1:289–311, 2015.

[101] Luyang Li, Yun-Hui Liu, Mu Fang, Zhizeng Zheng, and Hengbo Tang. Vision-based intelligent forklift automatic guided vehicle (agv). In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 264–265, 2015.

[102] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry : Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine*, 19(2):78–90, 2012.

[103] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[104] Swisslog. *https://www.swisslog.com/*, April 2022.

[105] Gaël Écorchard, Karel Košnar, and Libor Přeučil. Wearable camera-based human absolute localization in large warehouses. In Wolfgang Osten and Dmitry P. Nikolaev, editors, *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, pages 754 – 761. International Society for Optics and Photonics, SPIE, 2020.

[106] Mike Laughton. Open source data matrix software & library. https://github.com/dmtx/libdmtx.

[107] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[108] K Rainer, Giorgio Grisetti, and Kurt Konolige. g 2 o : A General Framework for Graph Optimization. pages 3607–3613, 2011.

[109] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34(5):827–828, Sep 1978.

[110] Gabriel Deak, Kevin Curran, and Joan Condell. A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16):1939–1954, 2012.

[111] J. González, J.L. Blanco, C. Galindo, A. Ortiz de Galisteo, J.A. Fernández-Madrigal, F.A. Moreno, and J.L. Martínez. Mobile robot localization based on ultra-wide-band ranging: A particle filter approach. *Robotics and Autonomous Systems*, 57(5):496–507, 2009.

[112] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems. *Procedia Computer Science*, 175:357–364, 2020.

[113] Sandra Djosic, Igor Stojanovic, Milica Jovanovic, Tatjana Nikolic, and Goran Lj. Djordjevic. Fingerprinting-assisted uwb-based localization technique for complex indoor environments. *Expert Systems with Applications*, 167:114188, 2021.

[114] Yang Song and Li-Ta Hsu. Tightly coupled integrated navigation system via factor graph for uav indoor localization. *Aerospace Science and Technology*, 108:106370, 2021.

[115] Yuan Xu, Choon Ki Ahn, Yuriy S. Shmaliy, Xiyuan Chen, and Yueyang Li. Adaptive robust ins/uwb-integrated human tracking using ufir filter bank. *Measurement*, 123:1–7, 2018.

[116] Christof Röhrig, Christopher Kirsch, J. Lategahn, Marcel Müller, and L. Telle. Localization of autonomous mobile robots in a cellular transport system. 2012.

[117] Stefania Monica and Gianluigi Ferrari. Low-complexity uwb-based collision avoidance system for automated guided vehicles. *ICT Express*, 2(2):53–56, 2016.

[118] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal on Robotics and Automation*, 3(3):239–248, 1987.

[119] Agostino Martinelli. Modeling and estimating the odometry error of a mobile robot. *IFAC Proceedings Volumes*, 2001. 5th IFAC Symposium on Nonlinear Control Systems 2001, St Petersburg, Russia, 2001.

[120] C. Golban, P. Cobarzan, and S. Nedevschi. Direct formulas for stereo-based visual odometry error modeling. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 197–202, 2015.

[121] J. Fabian and G. M. Clayton. Error analysis for visual odometry on indoor, wheeled mobile robots with 3-d sensors. *IEEE/ASME Transactions on Mechatronics*, 19(6):1896–1906, 2014.

[122] Andrea De Maio and Simon Lacroix. Simultaneously learning corrections and error models for geometry-based visual odometry methods. *IEEE Robotics and Automation Letters*, 5(4):6536–6543, 2020.

[123] Willy Hereman and S.W. Murphy Jr. Determination of a position in three dimensions using trilateration and approximate distances. *Department of Mathematical and Computer Science (MCS), Colorado School of Mines, USA*, 1995.

LIST OF FIGURES

Figure 2.1 Pinhole camera model. A light beam coming from a 3D point **X** goes through the optical center C and intersects an image plane in point **x**. 8

Figure 2.2 Calibration targets, checkerboard (upper) and aprilgrid (lower), recorded in several poses during the camera calibration process. 11

Figure 2.3 A stereo camera model: 3D points X_1-X_5 project to the same point u_l in the left image plane, but in the right image plane they project to 5 different points, u_{r1} - u_{r5}. 12

Figure 2.4 Comparison of epipolar lines before and after rectification. Before, the lines are slightly slanted (most obvious in top and bottom lines). After stereo rectification, the epipolar lines become horizontal. The points marked with red and green circles in one image have a match somewhere along the epipolar line in the other image. 13

Figure 2.5 A visualization of census transform. An intensity of a center pixel is a reference value, darker pixels become zeroes and brighter become ones. The neighborhood is then reshaped in a binary word. 15

Figure 2.6 Safety Vest with the sensor setup that consists of an IMU-aided stereo camera and a downward-looking monocular camera. This placement of the sensors was chosen since it will not disturb the human when performing the usual tasks. Furthermore, cameras cannot get obstructed by hands, and this part of the human body is the most stable and has the smallest chance of doing abrupt motion that could blur the images. 24

Figure 2.7 The fiducial marker used for localization of CarryPick robots in the Swisslog's warehouses. Each marker has a special ID and a different combination of the 9 rectangular patterns, so-called Data-Matrix. 24

Figure 2.8 RealSense D435 camera placed in between viewing areas of stereo camera and monocular camera to improve extrinsic calibration. Only a color camera was used during the calibration. 25

Figure 2.9 The experimental arena for collecting Dortmund dataset. The plastic boxes imitate racks and walls. The arena is covered with Optitrack motion capture system. (By courtesy of Fraunhofer IML). 25

Figure 2.10 The testing arena of Augsburg dataset. This arena is normally used to test algorithms on robots before the deployment, and its appearance (racks, ground markers, safety fence) is closely similar to the real warehouse. 26

Figure 2.11 AprilTags placed throughout the testing arena for acquiring the ground truth pose information. All AprilTags are unique and their map is computed with TagSLAM package. 26

Figure 2.12 The visual sensor suite, stereo and monocular camera, upgraded with the UWB sensor. 27

Figure 2.13 The floorplan of the faculty library in the Zagreb dataset. Blue squares mark the position of ground markers, the orange ellipsoid marks the area with presence of UWB anchors, and the green ellipsoid shows the area where the ground truth was available. 28

Figure 3.1 Fitting a quadratic curve on neighboring costs of a disparity d at a pixel $\mathbf{p}$. The minimum of a curve gives a subpixel disparity estimate d_{min} of pixel $\mathbf{p}$. 33

Figure 3.2 A scene recorded in two subsequent moments with a moving camera. Under the assumption that the environment is static, known pose change from X_{t-1} to X_t can be used to transform disparity D_{t-1} to D_t. 34

Figure 3.3 Block diagram of the proposed SGM algorithm. At the beginning, the classic SGM is computed in the full search space. Otherwise, the ego-motion estimation predicts the disparity map from the values of the previous step. The reduced disparity search space SGM block uses a stereo image pair and the disparity and variance prediction to compute the new disparity D_k^z. The computed disparity map D_k^z and the predictions $D_{k|k-1}$ and $P_{k|k-1}$ are used to correct the disparity map and variance. Outside the loop, the disparity is post-processed to improve the appearance of the final disparity map. 35

Figure 3.4 Top image: A computed disparity map D_{k-1} with 135th KITTI *Stereo 2015* benchmark pair. Bottom image: prediction $D_{k|k-1}$ based on visual odometry pose transformation. Some pixels are left without the predicted disparity because no pixel from the previous step maps to them, causing the zooming effect. 39

Figure 3.5 Histograms of disparity errors accumulated over the training sequences. The histograms are used to determine the process and measurement noise. 41

Figure 3.6 The visualization of disparity error accumulated over sequence 61. The intensity of each pixel represents the amount of the accumulated error. 41
Figure 3.7 Moving object detection on scenes 183, 45, and 46 in KITTI *scene flow 2015* benchmark. Top images in pair show detections in diff matrix image, while bottom images show detections in overlapped images from two consecutive time steps. 44
Figure 3.8 Comparison of outliers computed with OpenCV SGM and the proposed approach in the sequence *0051*. The peaks of the blue curve correspond to the frames with moving cars. 46
Figure 3.9 Images 100 , 254, 305, and 400 of sequence 51. The number of outliers has peaks in the images with a lot of moving objects. 46
Figure 3.10 Images from two consecutive time steps from KITTI sceneflow benchmark. 49
Figure 4.1 The concept of the proposed visual human localization system. 54
Figure 4.2 Matching ORB features in the downward-looking camera image to the reference marker image. 55
Figure 4.3 Steps of the marker-based pose estimation algorithm: a) ground marker in the original image, b) result of the morphological opening, c) image correlated with a double kernel, d) thresholded correlation image, e) marker with the computed orientation, f) the double kernel. 56
Figure 4.4 Two-dimensional visualization of an agent (black triangle) moving through an environment with features (red circles). Dashed poses are previous agent poses and the black curve connecting them represents the trajectory. When the agent is sufficiently close to a feature, sensors are able to measure the relative pose of the agent with respect to the feature (green dashed lines). 60
Figure 4.5 Construction of the pose graph with the pose nodes obtained from visual odometry (X_i) and the ground marker nodes (G_i). Green nodes and edges represent the output of the visual odometry algorithm, while orange nodes and edges represent the output of the marker detection algorithm. 63
Figure 4.6 The top-down view of trajectories for two sequences from the Dortmund dataset. Descrete jumps are caused by lack of ground truth data. 66
Figure 4.7 Comparison of the online (blue) and the offline (cyan) *orbslam2* trajectory with ground truth (purple) for sequence DM01. 67
Figure 4.8 The absolute position error in time of the proposed solution (blue) and multiple runs of ORB-SLAM2 on the sequence DM12345. The value −1 signifies that no pose estimates are produced by ORB-SLAM2 due to losing tracks of features. 68

Figure 4.9 Augsburg dataset - trajectory examples. The ground truth was not available along the whole trajectory, but only at the two sections marked with yellow circles. All sequences begin and end at the same position. 69

Figure 4.10 Comparison of the online (green) and the offline (blue) *fusion* trajectory on sequence AG03. The first marker detection, detected by discrete jumps in online trajectory, appeared after visiting the areas with ground truth marked with yellow circles. 71

Figure 4.11 The kidnapped-human scenario captured in four different time steps. Red squares represent the racks present in the testing arena. Blue and green trajectories with the corresponding pose arrows represent the *fusion* and the *odometry* estimations respectively. The red arrow represents the last detected pose with the ground stickers detection. With help from the sticker detection, the blue trajectory manages to return to the starting position, whereas the green trajectory drifts away. 73

Figure 4.12 Loss of localization in time for 5 different runs of ORB-SLAM2 and the proposed algorithm on the merged sequences AG02 and AG03. 74

Figure 5.1 Information flow in the proposed localization system. Red arrows show visual cues used for localization of the robots and the human worker. The *Location server* acts as an information broker and the worker localization algorithm can query the location of any robot at any time. 79

Figure 5.2 The graph constructed with visual sensors and UWB anchors network. The orange nodes represent the global pose estimates computed with the marker detection algorithm. The position of the blue node is computed with ranging UWB measurements through trilateration process, while the orientation is taken from the current orientation estimate. Green nodes represent the Safety Vest estimate and the green edges which connect them are computed with the visual odometry. 80

Figure 5.3 A simplified 2D example of error propagation through three consecutive steps. The initially introduced translation error (left) keeps the same magnitude in the following steps. The initial rotation error (right) remains the same rotation error magnitude, but the translation error increases with the distance from the initial step. 82

Figure 5.4 A plot of the function which maps the estimated error e_n magnitude into the edge weight $o^1_{n,n-1}$. The initial weight $O_{\max}$ linearly decreases until it reaches the minimum weight value $O_{\min}$. 83

Figure 5.5 A plot of the function which maps the average number of features per frame in the edge Φ to the edge weight $o^2_{n,n-1}$. The weight linearly increases with the number of features per frame between upper and lower limits $O_{\max}$ and $O_{\min}$. 83

Figure 5.6 Comparison of four trajectories computed with: fusion presented in the previous chapter, fusion with UWB location cues, fusion with UWB location cues and edge reweighting, and ORB-SLAM2 on sequences FL01, FL04, and FL10. 87

Figure 5.7 A setup for localization experiments with moving UWB anchors. Two moving UWB anchors are placed on the boxes which simulate moving robots. 88

LIST OF TABLES

Table 2.1 The list of dominant movements per recording in the Zagreb dataset. 29

Table 3.1 Sizes of sliding windows and corresponding thresholds. The threshold is a multiplication of window area and constant t which decreases from 1 in steps of 5%. The windows with diff value higher than the threshold are considered as potentially containing a moving object. 42

Table 3.2 The parameters used in the evaluation. 45

Table 3.3 OpenCV and proposed implementation comparison. Outliers are defined as on the KITTI benchmark (absolute threshold of 3 pixels and relative threshold of 5%). Diagonal and nondiagonal columns show the results computed using only 4 of 8 accumulations paths. 45

Table 3.4 The comparison of the best performing version of the proposed SGM approach with the top-ranking learning-based approach with open source code LEAStereo. The fourth and the fifth columns show the outlier percentage. Outliers are defined as on the KITTI benchmark (absolute threshold of 3 pixels and relative threshold of 5%). 47

Table 4.1 The offline trajectory results for the Dortmund dataset. The first three rows show the absolute trajectory error in meters for each sequence, the fourth row shows the total distance traveled during the recording, and the last row contains the number of detected ground markers (note that 2 markers are always detected at the start and end of sequence). 65

Table 4.2 The online trajectory results for the Dortmund dataset. The table shows the absolute trajectory error in meters for each sequence. 67

Table 4.3 The results for the Augsburg dataset. The second, third and fourth columns show the absolute trajectory error in meters, the fifth column shows the total distance traveled for each of the recordings in meters, and the last column is the number of detected markers for the sequence. 70

Table 4.4 The online trajectory results for the Augsburg dataset. The rows show the absolute trajectory error in meters for each sequence. 71
Table 4.5 The localization gap intervals and the distances between poses of the *fusion* trajectory at the beginning and the end of each interval. 74
Table 5.1 The values of parameters used for evaluation. 84
Table 5.2 The results for the *Augsburg* dataset. All values are expressed in meters. 85
Table 5.3 The results for the *Dortmund* dataset. All values are expressed in meters. 85
Table 5.4 The results for the *Zagreb* dataset. All values are expressed in meters. 86
Table 5.5 Dynamic UWBs localization experiment. All values are expressed in meters. 88

CURRICULUM VITAE

Goran Popović was born in Zagreb, Croatia in 1992. He graduated from the mathematically oriented high school in 2011 at the V Gymnasium in Zagreb. He received his mag. ing. degree (cum laude) from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER) in 2017. He finished the first semester of his master program at the Vienna University of Technology, Austria, as an exchange student. Between the first and the second year of his graduate studies, he completed an internship at Bosch, Abstatt (2016) within the framework of the Foundation "Internship Programme of German Business for the Countries of the Western Balkans".

During his undergraduate studies he received a scholarship from the Foundation "Prof. Zlata Bartl" (2013), a scholarship from the National Foundation for the Support of Student Living Standard (2014), a DAAD scholarship for a German language course in Passau, Germany (2014) and the Rector's Award (2015). During his graduate studies he was awarded the scholarship of the City of Zagreb (2015-2017) and the bronze plaque "Josip Lončar" (2017).

Since October 2017, he has been working as a research assistant in the Department of Control and Computer Engineering at FER in Zagreb. Since then, he has been participating in the international Horizon 2020 scientific project Safe Human-Robot Interaction in Logistic Applications for Highly Flexible Warehouses - SafeLog. In 2021, he was engaged in the project Development of Environmentally Friendly Vehicle for Cleaning Public Surfaces with Autonomous Control System Based on Artificial Intelligence - EKO-KOMVOZ, funded by the European Regional Development Fund. In 2018, he was a visiting researcher at the University of Karlsruhe for one month.

As a result of his research, he published one journal paper and three conference papers. His main research area is visual perception with focus on localization with visual sensors.

PUBLICATIONS

JOURNAL PUBLICATIONS:

1. G. Popović, I. Cvišić, G. Écorchard, I. Marković, L. Přeučil and I. Petrović. Human localization in robotized warehouses based on stereo odometry and ground-marker fusion. *Robotics and Computer-Integrated Manufacturing*, 73:102241,2022.

CONFERENCE PUBLICATIONS:

1. G. Popović, J. Oršulić, D. Miklić and S. Bogdan. Rao-Blackwellized Particle Filter SLAM with Prior Map: An Experimental Evaluation. *Third Iberian Robotics Conference (Robot2017)* Sevilla, Spain, pages 14–25, 2017.

2. G. Popović, A. Hadviger, I. Marković and I. Petrović. Computationally efficient dense moving object detection based on reduced space disparity estimation. *12th IFAC Symposium on Robot Control (SYROCO 2018)* Budapest, Hungary, pages 360–365, 2018.

3. G. Popović, I. Cvišić, I. Marković and I. Petrović. Human Localization in Warehouse Environments based on a Wearable Camera Sensor Suite and Dynamic Ultra-Wide Band Nodes. *IEEE 20th International Conference on Advanced Robotics (ICAR)* Ljubljana, Slovenia, pages 818–823, 2021.

ŽIVOTOPIS

Goran Popović rođen je u Zagrebu, Hrvatska, 1992. godine. Prirodoslovno-matematički orijentiranu V. gimnaziju završio je 2011. godine u Zagrebu. Akademski naziv mag. ing. elektrotehnike i informacijske tehnologije (cum laude) stekao je 2017. godine završivši diplomski studij u polju elektrotehnike na Sveučilištu u Zagrebu, Fakultetu elektrotehnike i računarstva (FER). Prvi semestar diplomskog studija završio je na Tehničkom sveučilištu u Beču, Austrija, u sklopu studentske razmjene. Između prve i druge godine diplomskog studija pohađao je ljetnu praksu u kompaniji Bosch, Abstatt (2016.) u sklopu udruge "Internship Programme of German Business for the Countries of the Western Balkans".

Tijekom preddiplomskog studija nagrađen je stipendijom udruge "Prof. Zlata Bartl" (2013.), stipendijom Nacionalne zaklade za potporu učeničkom i studentskom standardu (2014.), stipendijom udruge DAAD za školu njemačkog jezika u Passau, Njemačka (2014.), te rektorovom nagradom (2015.). Tijekom diplomskog studija nagrađen je stipendijom Grada Zagreba (2015.-2017.) i brončanom plaketom "Josip Lončar" (2017.).

Nakon diplomskog studija zaposlen je na mjestu asistenta na Zavodu za automatiku i računalno inženjerstvo na FER-u. Od 2017. do 2021. godine sudjeluje na međunarodnom znanstvenom projektu iz Obzora 2020 Sigurna čovjek-robot interakcija u logističkim primjenama za visoko fleksibilna skladišta (SafeLog). Od 2021. sudjeluje u projektu Ekološki prihvatljivo vozilo za čišćenje javnih površina sa sustavima autonomnog upravljanja zasnovanim na umjetnoj inteligenciji (EKO-KOMVOZ), financiran Europskim fondom za regionalni razvoj. Tijekom 2018. godine proveo je jedan mjesec kao gostujući istraživač na Tehničkom institutu u Karlsruheu, Njemačka.

Njegovo glavno područje interesa je vizualna percepcija s fokusom na lokalizaciju vizualnim senzorima, a rezultate svojih istraživanja objavio je u jednom časopisnom i tri konferencijska znanstvena rada.

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.