



Förstudie: Robust kartering och positionsbestämning med soldatburna sensorer

Sensorsystem för urban miljö: Red & Blue Force Tracking

THORD ANDERSSON

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.



FOI
Totalförsvarets forskningsinstitut
Sensorsystem
Box 1165
581 11 Linköping

Tel: 013-37 80 00
Fax: 013-37 81 00

www.foi.se

FOI-R--2525--SE Teknisk rapport
ISSN 1650-1942 Juni 2008

Sensorsystem

Thord Andersson

Förstudie: Robust kartering och positionsbestämning med soldatburna sensorer

Sensorsystem för urban miljö: Red & Blue Force Tracking

Titel	Förstudie: Robust kartering och positionsbestämning med soldatburna sensorer
Title	Feasability Study: Robust Mapping and Localization with Soldier Wearable Sensors
Rapportnr/Report no	FOI-R--2525--SE
Rapporttyp Report Type	Teknisk rapport Technical report
Sidor/Pages	29 P
Månad/Month	Juni / June
Utgivningsår/Year	2008
ISSN	ISSN 1650-1942
Kund/Customer	Försvarsmakten(FM)
Forskningsområde Programme area	4. Sensorer och signaturanpassning 4. Sensors and Low Observables
Delområde Subcategory	42 Sensorer 42 Above surface Surveillance, Target acquisition and Reconnaissance
Projektnr/Project no	E3087
Godkänd av/Approved by	Jörgen Ahlberg
FOI, Totalförsvarets Forskningsinstitut	FOI , Swedish Defence Research Agency
Avdelningen för Sensorsystem	Sensor Systems
Box 1165	P.O.Box 1165
581 11 Linköping	SE-581 11 Linköping

Sammanfattning

Vi vill ge den enskilde soldaten och stridsgruppen möjlighet att gemensamt, distribuerat och fullständigt automatiskt, kartera sin omgivning *och samtidigt* positionsbestämma sig relativt den uppbyggda kartan. Dessa nya förmågor skulle förbättra beslutsstödet och situationsuppfattningen för den enskilde soldaten och stridsgruppen vid snabba dynamiska insatser i byggnader, eller andra områden med begränsat externt stöd, t.ex. där GPS eller andra lokaliseringsmöjligheter saknas eller bara fungerar sporadiskt.

Denna förstudie diskuterar hur SLAM, en metod från robotiken, kan användas tillsammans med enkla bärbara sensorer och kommunikationsutrustning för samtidig kartering och lokalisering. Automatisk följning av rörliga objekt i stridsgruppens omgivning diskuteras också.

Nyckelord: Lokalisering, Positionsbestämning, Kartering, SLAM, Följning, Datafusion

Summary

We want to enable a small number of soldiers to automatically map their surroundings while simultaneously estimating their positions relative this map. These new abilities would enhance the situational awareness of the soldiers in situations where GPS or other localization means are not available or dependable, for instance when entering an urban building.

This feasibility study show how SLAM, a method from robotics, can be used together with simple wearable sensors and communication equipment for simultaneous localization and mapping. In addition, automatic tracking of moving objects in the vicinity of the soldiers is discussed.

Keywords: Localization, Mapping, SLAM, Tracking, Data Fusion

Innehållsförteckning

1	Inledning	7
1.1	Scenario och önskade förmågor	7
1.2	Systemkrav	8
2	Samtidig lokalisering och kartläggning	9
2.1	Inledning.....	9
2.2	Bakgrund.....	9
2.3	Definition	10
2.4	Delproblem.....	12
2.4.1	Datafusion	13
2.4.2	Dataregistrering	13
2.4.3	Dataassociation	14
2.4.4	Tillståndsestimering	14
2.4.5	Klassificering / Identifiering	15
2.5	Vanliga SLAM-metoder	16
2.5.1	EKF-SLAM	16
2.5.2	FastSLAM	16
2.6	Forskningsfronten i SLAM.....	17
2.6.1	Beräkningskomplexitet.....	17
2.6.2	Dataassociation	17
2.6.3	Omgivningsrepresentation	18
2.6.4	Multirobotsystem.....	19
2.7	Exempel på implementerade system.....	20
2.7.1	MonoSLAM	20
2.7.2	Newman et al., "Explore and Return"	20
2.7.3	Guivant och Nebot, SLAM i stora omgivningar utomhus.....	20
2.7.4	Consistent Pose Estimation	21
3	Diskussion och förslag till fortsatt arbete	22
3.1	Uppfyllelse av tekniska krav.....	22
3.1.1	Sammanfattning.....	23
3.2	Förslag till fortsatt arbete	23
3.2.1	Prototypsystem	23
3.2.2	Forskningsverktyg.....	24
4	Referenser	25

1 Inledning

Den vision för den framtida soldaten som har uppmålats i projekt som MARKUS [1], brittiska arméns FIST (Future Integrated Soldier Technology) [2], tyska IdZ-ES (Infanterist der Zukunft) [3] och amerikanska arméns Land Warrior [4] börjar i och med Irak-krigets behov att infrias. Gemensamt är att den enskilde soldaten är utrustad med en rad sensorer, datorutrustning och avancerad kommunikationsutrustning. Storbritannien har bland annat inlett anskaffning av system för det man kallar ELSA (Enhanced Local Situation Awareness) [5], vilket innefattar mjukvarudefinierade radiosystem (SDR, Software Defined Radio), handhållna terminaler och GPS-lokalisering av egen trupp. På terminalerna kan exempelvis viss ordergivning göras, och kartor kan visas där t.ex. egna soldaters position markeras tillsammans med indikationer för viktiga händelser. Specialtrupper använder dessutom redan nu ljudförbättrings- och ljudskyddssystem med integrerad kommunikation, hjälmkameror, IR/video-sikten m.m. Vidare studeras portabla väggpenetrerande radarsystem och avbildande lasersystem med stort intresse av många parter, däribland FOI.

I denna rapport kommer vi att undersöka hur man kan utnyttja de enskilda soldaternas sensorer för att positionsbestämma soldaterna och *samtidigt* kartera omgivningen i områden med begränsat yttre stöd (ingen GPS, bristfälliga kartor etc.), t.ex. vid genomsökning av ett okänt hus. Detta är en viktig delförmåga inom ”Red and Blue Force Tracking”.

Vi kommer först att definiera problemet med hjälp av ett tänkt scenario i avsnitt 1.1, för att sedan diskutera vilka krav en teknisk lösning måste uppfylla i avsnitt 1.2.

I kapitel 2 kommer vi sedan att diskutera en metod för samtidig positionsbestämning och kartering som kallas SLAM (Simultaneous Localization And Mapping), och hur den kan användas i denna tillämpning. SLAM har en oerhörd potential att förbättra beslutsstöd och situationsförståelse för den enskilda soldaten eller stridsgruppen i situationer där externt stöd saknas.

Avslutningsvis kommer vi i kapitel 3 presentera ett prototypsystem som har potential att uppfylla de uppställda kraven. Vi kommer vidare att sammanfatta rapporten samt ge rekommendationer till fortsatt arbete.

1.1 Scenario och önskade förmågor

Ett antal egna soldater (blue force) närmar sig ett okänt hus med okänt innehåll. Man befärrar att fientliga styrkor (red force) kan finnas i huset. Förmodligen finns där även civila, och eventuellt även neutrala styrkor. Uppgiften för soldaterna är att kartlägga huset och bekämpa eventuellt motstånd. Vi antar att soldaterna har ett antal relativt enkla och billiga sensorer integrerade i sin utrustning, t.ex. i uniformen, hjälmen, sikten, vapen osv. Dessutom antar vi att soldaterna har en integrerad kommunikationsutrustning som tillåter datatrafik och dessutom anpassar sig efter kommunikationsavstånd, störningar, bandbreddsbehov osv. Det finns ingen möjlighet till GPS-positionering i huset och det finns normalt ingen förhandsinformation om hur huset ser ut inomhus.

Vi vill ge den enskilde soldaten och stridsgruppen möjlighet att gemensamt, distribuerat och fullständigt automatiskt, kartera sin omgivning *och samtidigt* positionsbestämma sig relativt den uppbyggda kartan. Dessa nya förmågor skulle förbättra beslutsstödet och situationsuppfattningen för den enskilde soldaten och stridsgruppen vid snabba dynamiska insatser i byggnader, eller andra områden med begränsat externt stöd (t.ex. där GPS, kommunikation och kartmaterial saknas eller bara fungerar sporadiskt).

1.2 Systemkrav

Utifrån scenariet i avsnitt 1.1 kan vi definiera följande krav som en teknisk lösning måste uppfylla för att vara praktiskt intressant:

1. *Sensorbörda*: Sensorerna, skärm/terminal, kommunikationsutrustning och annat som soldaten bär eller har på sig måste naturligtvis vara lätt och bekvämt att bära, vara robust och dra lite ström. Vidare får utrustningen inte röja soldaten för fientlig utrustning på något trivialt sätt.
2. *Distribuerad funktionalitet*: Viktigt är också att positionsbestämningen och karteringen är redundant, robust och distribuerad. Om något händer en enskild soldat så skall inte den övriga stridsgruppen tappa funktionalitet.
3. *Realtidskrav*: Systemet måste klara av dynamiska och föränderliga omgivningar där mycket kan hända snabbt.
4. *Dynamiska omgivningar*: Systemet måste kunna skilja på rörliga mål (t.ex. människor), stationär omgivning (t.ex. väggar, fixerade möbler) och semistationära objekt (t.ex. dörrar, stolar o.dyl.)
5. *Resursbudget*: Algoritmerna måste vara mycket beräkningseffektiva för att minimera strömförbrukningen och undvika dyr specialhårdvara. Vidare får de bara utnyttja kommunikationsutrustningen i mån av tillgänglig bandbredd, vilket kräver en inbyggd skalbarhet i detta avseende.
6. *Kartkvalitet*: Kvaliteten och utformningen hos kartorna skall i första hand vara anpassad till soldatens orienterings- och situationsuppfattningsbehov. I andra hand, och i mån av beräknings- och kommunikationsresurser, kan mer fullödiga kartor beräknas (3d, objekt, texturer o.dyl.).
7. *A priori information*: A priori information (t.ex. partiella kartor) måste kunna hanteras och utnyttjas av systemet.
8. *Okänd initialposition*: För att kunna hantera utrustningsproblem måste systemet klara av en okänd initialposition och, speciellt, okända relativa positioner mellan soldaterna. Normalt kommer dock troligen en initialposition vara känd för varje soldat då man inträder byggnaden (bestämd med t.ex. GPS).

2 Samtidig lokalisering och kartläggning

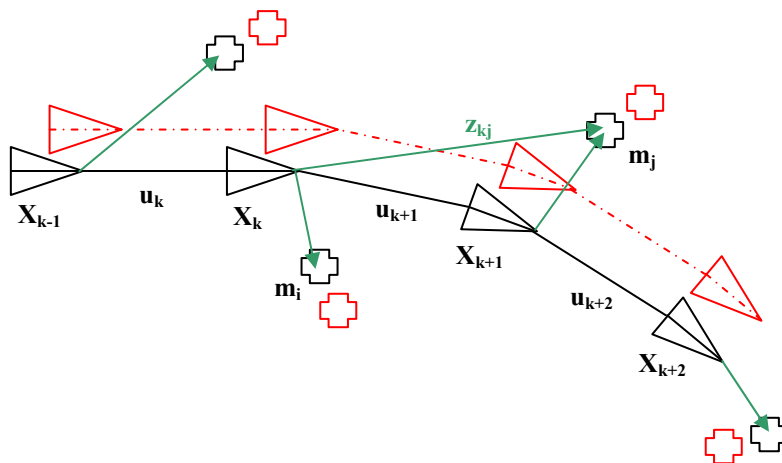
2.1 Inledning

Samtidig lokalisering och kartläggning har studerats länge inom robotiken under namnet *SLAM* (*Simultaneous Localization And Mapping*) [6, 7, 8, 9]. SLAM har genomgått en oerhörd utveckling under de senaste 10 åren och har börjat användas inom allt fler områden då tekniken har börjat mogna och beräkningsresurserna har ökat. Dock återstår en del problem att lösa.

I detta kapitel kommer vi först att definiera det grundläggande SLAM-problemet och hur forskningsfältet har utvecklats sedan 1980-talet. Sedan kommer vi att identifiera några viktiga delproblem och presentera en översikt¹ över dessa och vanliga lösningsmetoder. En beskrivning¹ av den nuvarande forskningsfronten kommer i avsnitt 2.5 samt 2.6 där de mest framgångsrika SLAM-varianterna av idag presenteras och deras för- och nackdelar diskuteras. I avsnitt 2.6 diskuteras också olika typer av problem som de nuvarande SLAM-varianterna uppvisar och hur man kan överkomma dessa problem. Kapitlet avslutas med en diskussion kring hur SLAM-metoderna passar till det problem vi presenterade i kapitel 1; vad finns, vad är på gång och vad fattas?

2.2 Bakgrund

I SLAM-problemet försöker man besvara frågan om det är möjligt för en mobil robot att, utifrån en okänd position i en okänd miljö, stegvis bygga upp en konsistent karta av omgivningen *samtidigt* som den bestämmer sin position relativt den kartan. I Figur 1 ser vi hur en robot rör sig framåt. Positionsestimaten är rödmarkerade och den sanna positionen svart. Från varje position kan roboten göra en mätning (grönmarkerat) mot ett eller flera landmärken. Ett landmärke är någon robust detekterbar egenhet hos omgivningen som kan variera från sensortyp till sensortyp. Exempel på estimerade kartor visas i Figur 2 nedan.



Figur 1: En robot har posen (positionen och orienteringen) X_k och observerar därifrån landmärket m_i och m_j . Efter en styrsignal u_{k+1} har roboten tillståndet (eller posen) X_{k+1} och observerar därifrån m_j . I figuren är de sanna positionerna svarta och de estimerade röda. Mätningarna är markerade med gröna pilar.

¹ Avsnitten är baserade på de utmärkta översikterna i [6,7,8,16]. Den intresserade läsaren rekommenderas starkt att läsa dessa för en mer detaljerad och ingående presentation av datafusion, SLAM och dess varianter.

SLAM-problemet har visat sig vara teoretiskt och konceptuellt lösbart vilket är en av de största framgångarna inom robotiken. Problem återstår däremot för hur man i praktiken skall implementera mer generell tillämpbara lösningar och hur man skall bygga mer visuellt avancerade kartor. Vi återkommer till nuvarande begränsningar senare.

1986 kan sägas vara starten på den moderna SLAM-forskningen då en rad forskare samlades på IEEE Robotics and Automation Conference och insåg att SLAM utgjorde ett fundamentalt och viktigt problem inom många områden [6]. De första artiklarna som därefter utkom, behandlade problemet med hur man skall beskriva relationer mellan landmärken och hur de geometriska osäkerheterna skall representeras.

1990 kom en viktig artikel [10] som visade att när en mobil robot rör sig genom en omgivning och gör relativa mätningar av landmärkena, så kommer positionsestimaten för dessa landmärken bli korrelerade med varandra på grund av det gemensamma felet i robotens positionsestimat. Detta innebär att när man får in en ny mätning av ett landmärke så måste estimaten för *alla* landmärken uppdateras tillsammans med robotens positionsestimat. Då beräkningskomplexiteten för detta är hög, ungefär kvadraten på antalet landmärken, och då man trodde att felet hos kartestimatet inte skulle konvergera, försökte man istället hitta approximationer till problemet. Med approximationerna försökte man minimera korrelationerna mellan landmärkena, och på så sätt faktorisera SLAM-problemet till ett separat lokaliserings- och ett separat karteringsproblem.

Runt 1995 kom ett genombrott [11] då man, tvärt emot den tidigare uppfattningen, insåg och kunde visa att felet hos kartestimatet verkligen konvergerade om man formulerade SLAM som ett enda estimeringsproblem, och att lösningen blev bättre och bättre desto större korrelationen mellan landmärkena var. Faktum är att korrelationerna mellan landmärkena ökar monotont med antalet observationer. Detta innebär t.ex. att estimatet för ett landmärke \mathbf{m}_1 blir bättre om roboten senare observerar ett landmärke \mathbf{m}_2 , *även om \mathbf{m}_1 inte syns från den nya positionen!* Alla landmärken är därför länkade till varandra och till roboten. Man kan visualisera detta som om alla landmärken och roboten är kopplade till varandra med fjädrar vars styvhet är analogt med korrelationerna [6].

Sommarskolan i SLAM 2002, som hölls i KTHs regi i Stockholm, blev en stor framgång för spridningen av fältet som sedan vuxit explosionsartat de senaste åren.

2.3 Definition

I detta avsnitt skall vi definiera hur SLAM är uppbyggt. Enligt Figur 1 kan vi definiera följande symboler:

- \mathbf{x}_k : En tillståndsvektor som beskriver positionen och orienteringen ("posen") hos roboten. Fortsättningsvis kommer vi att skriva "tillstånd" eller "pose" då vi refererar till \mathbf{x}_k i rapporten.
- \mathbf{u}_k : En styrsignalvektor som verkar på roboten vid tidpunkten $k-1$ så att den hamnar i läget \mathbf{x}_k vid tidpunkten k .
- \mathbf{m}_i : En vektor som beskriver det i :e landmärkets position. Man brukar anta att denna är tidsinvariant (vilket är en stor begränsning i den aktuella applikationen).
- \mathbf{z}_{kj} : En observation som är resultatet av en mätning på landmärket j vid tillfället k .
- $\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$. Robotens historiska bana, dvs. samtliga poser.
- $\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$. Samtliga styrsignaler genom historien.
- $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$. Mängden av alla landmärken.
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$. Mängden av alla observationer från landmärken.

I Bayesiansk online-SLAM söker vi:

$$\bullet \quad P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \quad (1)$$

dvs. sannolikhetsfördelningen för robotens tillstånd (position och orientering) och karta för alla tidpunkter k , givet alla observationer, styrsignaler och a priori-fördelningen för posen.

Normalt sett vill vi inte lagra på oss hela historien av observationer och styrsignaler utan hitta en rekursiv lösning för tidpunkten k givet lösningen för tidpunkten $k-1$. Vi behöver då en mätmodell som beskriver sannolikheten för en observation givet tillståndet och kartan:

$$\bullet \quad P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \quad (2)$$

samt en processmodell som beskriver dynamiken hos roboten:

$$\bullet \quad P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \quad (3)$$

Genom att använda dessa modeller kan vi nu implementera en SLAM-algoritm rekursivt utgående ifrån \mathbf{x}_0 a priori fördelningen:

Tidsuppdatering (Prediktion)

$$\begin{aligned} & P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) \\ &= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \end{aligned} \quad (4)$$

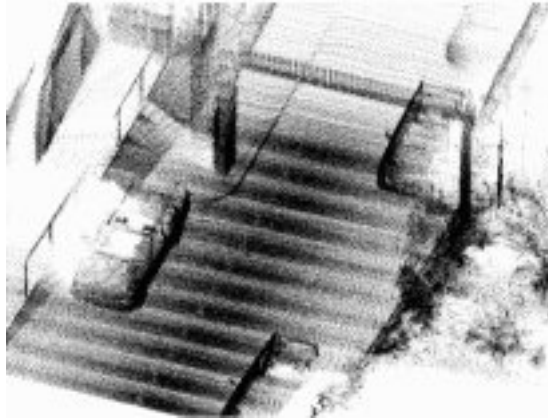
Mätuppdatering (Korrigerig)

$$\begin{aligned} & P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \\ &= \frac{P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})} \end{aligned} \quad (5)$$

Genom att iterera följande tvåstegsförfarande:

- först en prediktion av nästa tillstånd med ekvation (4)
- och sedan en korrigerig av prediktionen med nya observationer och ekvation (5)

får vi en ständig tillgång de sökta estimaten enligt ekvation (1). Exempel på estimerade kartor visas nedan i Figur 2.



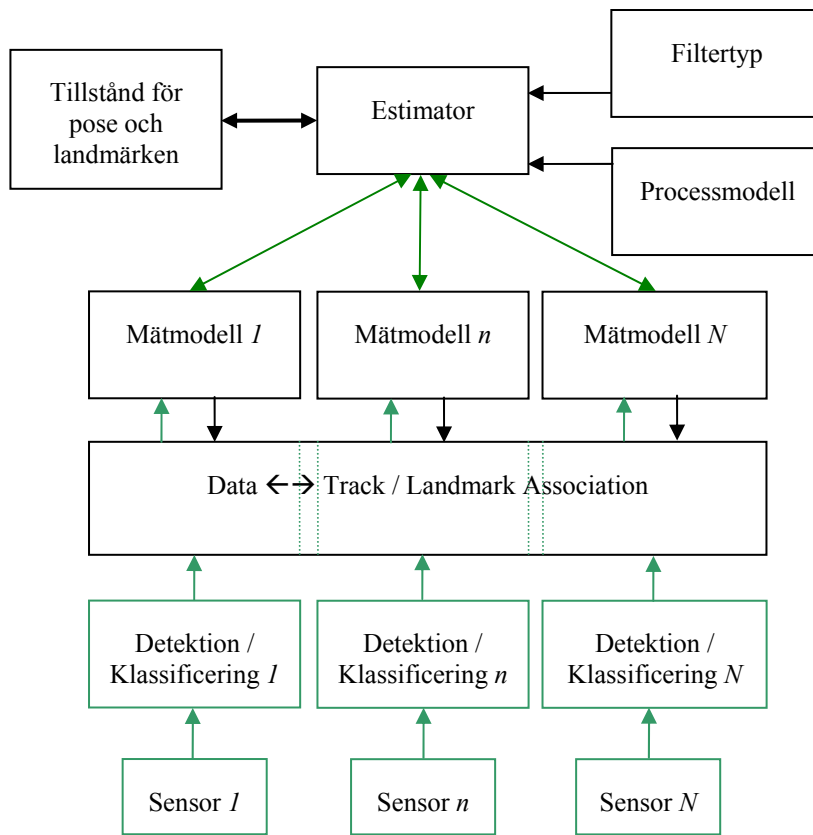
Figur 2: Överst en 3d-modell byggd med SLAM och 3d-laserdata. Underst en 2d-karta över en byggnad framtagen med SLAM. Bilderna kommer från Newman et al. [12]

2.4 Delproblem

I Figur 3 ser vi de olika beståndsdelarna i en SLAM-metod. Man kan observera att en målföljare med många sensorer och för många mål (MultiSensor MultiTarget Tracker) har samma uppbyggnad. Gemensamt är att man använder datafusionsmetoder för att kombinera data/information från många, ofta olika, sensorer för att erhålla säkrare utsagor om t.ex. ett objekts position och rörelse (följning) [13, 14, 15, 16].

Hur väl hela systemet fungerar är beroende av varje komponents prestanda. Nedan följer därför en kort översikt² över olika metoder för identifiering, dataassociation samt estimering. ”Landmärken”, ”målspar” eller ”spår” används synonymt i detta avsnitt.

² Avsnittet är baserat på [16] vilken rekommenderas för vidare läsning.



Figur 3: Figuren visar de olika delmodulerna i en SLAM-algoritm för många sensorer.

2.4.1 Datafusion

Datafusionsmetoder används för att kombinera data/information från många, ofta olika, sensorer för att erhålla säkrare utsagor om t.ex. ett objekts position och rörelse (följning) [17, 18, 19, 20]. Det finns många olika föreslagna fusionsarkitekturer men den vanligaste är JDL-modellen [21] som delar upp fusionsprocessen i fyra delar, Object-, Situation-, Threat- och Process Refinement. Endast den första presenteras här då de andra delarna inte är direkt relevanta för denna rapport.

JDL-nivå 1 kallas ofta datafusion, medan högre nivåer kallas informationsfusion. Distinktionen mellan data- och informationsfusion är vag, men i den senare används ofta mer symboliska och subjektiva metoder [22].

Object Refinement, JDL nivå 1, har som målsättning att kombinera sensordata så att optimalt noggranna och robusta estimat av ett objekts position, kinematik, signatur och identitet erhålls. Nivån delas algoritmiskt normalt upp [23] i dataregistrering, dataassociation, tillståndsestimering och identifiering, men andra varianter förekommer.

2.4.2 Dataregistrering

Behandlar data så att en gemensam referensram erhålls. Innefattar t.ex. koordinat-transformationer.

2.4.3 Dataassociation

Behandlar problemet hur man skall associera inkommande observationer till befintliga/nya målspar / landmärken. I distribuerade system används olika uppsättningar av målspar / landmärken och då måste dessa kunna associeras till varandra. Exempel på metoder är:

- *Nearest/All Neighbor*: Mycket enkla och beräkningseffektiva metoder som väljer den närmaste eller de närmaste observationerna för att uppdatera ett visst målspar. Vid många mål, äkta eller falska (klotter), uppvisar dessa metoder tyvärr stora fel [24].
- *Joint Probabilistic Data Association (JPDA)*: A posteriori sannolikheterna för olika kombinationer av observationer och målspar beräknas och en viktad tilldelning av observationer till målspar görs [25]. JPDA i sin originalformulering hanterar klotter på ett bra sätt men har problem med skapandet av nya målspar respektive terminering av gamla inaktuella målspar [26]. Ett antal varianter och vidareutvecklingar av JPDA har därför gjorts genom åren [26,27,28,29,30].
- *Lagrangian Relaxation*: Att optimalt lösa associationsproblemet kan visas vara NP³-svårt, då många sensorer används för att följa många mål. Lagrangian Relaxation är en teknik (av många) som istället effektivt hittar en nästan optimal lösning [31, 32].
- *Artificial Neural Networks (ANN)*: Hopfield-nätverk har med framgång använts till att approximativt lösa målspar till målspar associering, som också är NP³-svårt [33].
- *Fuzzy Logic*: Ett problem med JPDA-liknande metoder är att beräkningstiden ökar exponentiellt med antalet mål. Fuzzy Logic och Fuzzy Data Association angriper detta genom att diskretisera data till fuzzy sets [34,35,36].

2.4.4 Tillståndsestimering

Tillståndsestimering används för att t.ex. uppdatera ett landmärkes position med nya (associerade) observationer. Exempel på metoder är:

- *Kalmanfilter (KF/EKF)*: Kalmanfiltret [37] är i särklass den mest använda tekniken för tillståndsestimering för målföljning, känd som den optimala estimatoren för linjära gaussiska system. Kalmanfiltret tar in en sekvens av brusiga mätobservationer och, givet en process- och mätmodell, genererar en sekvens av tillståndsestimat, t.ex. position. För att bättre klara av olinjära system har KF vidareutvecklats till bl.a. Extended Kalman Filter (EKF) [38] och Sigma-Point Kalman Filter [39], som också innefattar Unscented Kalman Filter (UKF) [40].
- *Multiple Model Algorithms*: För att kunna följa manövrerande mål krävs det att filtrets dynamik överensstämmer med det verkliga systemet, annars divergerar filtret. De två vanligaste sätten att hantera detta är med Generalized Pseudo-Bayesian (GPB) [41, 42] och Interacting Multiple Model (IMM) [43] som båda använder filterbanker för att anpassa sig till målets dynamik. Båda dessa tekniker har fördelen av att vara modulära; filterbankerna kan innehålla filter såsom t.ex. KF, EKF eller JPDA.
- *Multiple Resolutional Filtering*: För att öka beräkningseffektiviteten och flexibiliteten hos trackingsystem har upplösningshierarkier använts där olika trackingtekniker kan

³ Ett problem tillhör klassen NP (*Nondeterministic Polynomial time*) om det är lösbart inom polynomisk tid med en icke-deterministisk Turing-maskin [32].

användas på olika upplösningsnivåer [44]. Genom att gå till högre upplösningsnivåer, och använda mer avancerade trackingmetoder bara när det verkligen behövs, kan stora prestandavinsterna göras. Även kvaliteten hos följningen kan bli bättre med denna teknik, t.ex. för svaga mål eller vid höga nivåer av bakgrundsbrus [45,46].

- *Particle Filter (PF)*: För att kunna hantera olinjära modeller och icke-gaussiska fördelningar bättre än t.ex. EKF, uppfanns partikelfiltret som representerar sannolikhetsdistributionen med ett antal rörliga partiklar [47]. PF är också känt som Bootstrap-, Condensation- eller Monte-Carlo filter. En noggrann beskrivning av olika PF hittas i [48].
- *AI-ansatser*: Normalt används Kalmanfilter eller Bayesianska tekniker när den statistiska modellen för t.ex. mätbrus är känd. Annars är vi hänvisade till fusion med hjälp av AI-tekniker såsom t.ex. regelbaserad fusion, artificiella neurala nätverk eller fuzzy logic:
 - *Regelbaserad*: En enkel approach till multisensor datafusion som använder en mängd heuristiska regler. Trots enkelheten kan approachen vara effektiv och används som baseline vid prestandajämförelser med andra AI fusionstekniker [49].
 - *Artificiella Neurala Nätverk (ANN)*: Många har framgångsrikt använt neurala nätverk för olika former av datafusion, inklusive tillståndsestimering, se [16] för referenser. Nyligen har också genetiska algoritmer (GA) använts för design av ANN för datafusion [50]. Support Vector Machines (SVM) bör också nämnas i sammanhanget som ett modernt alternativ [51].
 - *Fuzzy Logic*: Kalmanfiltret antar att process- och mätbruskovarianser är kända a priori. I verkliga processer måste dessa estimeras och kvalitén på dessa estimat har stor inverkan på filtrets prestanda och robusthet. [52] föreslår ett adaptivt Kalman Filter (FL-AKF) som använder sig av fuzzy logic för att justera kovariansmatriserna. Metoden har fått flera uppföljare [53]. Metoderna används för bl.a. feldiagnos för sensorer och scenarier med heterogena sensorer.

2.4.5 Klassificering / Identifiering

Om varje sensor utför klassificering och ger ett mål en identitet med tillhörande konfidensnivå, kan man fusionera de lokala utsagorna med olika tekniker till en global identitetsutsaga, s.k. beslutsfusion. Några vanliga metoder för detta följer nedan. Ett alternativ är att istället först fusionera datat från de olika sensorerna, och sedan klassificera det fusionerade resultatet (modellen).

- *Bayesian Inference (BI)*: Använder sig av statistisk slutledning med Bayes teorem.
- *Dempster-Shafer (D-S)*: Dempster-Shafer [54,55] generaliserade den traditionella Bayesianska "belief"-modellen till att tillåta explicita representationer av osäkerhet. D-S utsagor kan dock vara ointuitiva och problematiska att tolka [16]. En omfattande lista över D-S och Bayesianska algoritmer finns i [56].
- *Artificial Neural Networks (ANN)*: Återkopplade ANN såsom Hopfield-nätverk fungerar som ett associativt minne; Om en del av ett minne presenteras så återskapar nätverket det fullständiga minnet som ligger "närmast". Denna egenskap kan framgångsrikt utnyttjas för identifiering [57,58].
- *Expertsystem*: Kittler [59] föreslår att lokala expertsystem används för identifiering vid varje sensor. Ett viktat medelvärde av dessa beslut görs sedan för att fusionera

utsagorna.

- *Omröstnings- och summeringsansatser*: Dessa ansatser är populära och mycket enkla [60]. En bank av klassificerare används för varje sensor. Vid summeringsfusion adderas konfidensvärdet från varje klassificerare för varje hypotes. Den hypotes som har högst summa används. Vid omröstningsfusion används den hypotes som anses troligast av flest sensorer. Metoderna har använts i många sammanhang, t.ex. detektion av landminor och målföljning [16].
- *Distribuerad klassificering*: När man skall göra en distribuerad klassificering måste man bestämma sig för om man skall klassificera på fusionerat data eller fusionera lokala klassifikationer. Brooks et al. [61] argumenterar för att datafusion är överlägset när data är korrelerat medan beslutsfusion är bättre om data är okorrelerat. I [62] demonstreras vidare att beslutsfusion fungerar bra så länge data är felfritt, men att prestanda minskar snabbare än för datafusion då mätfel introduceras i systemet.

2.5 Vanliga SLAM-metoder

Genom att välja lämpliga representationer av mätmodellen (ekv. 2) och processmodellen (ekv. 3) så att beräkningen av priori- och posteriorifördelningen i tidsuppdateringen (ekv. 4) och mätuppdateringen (ekv. 5) blir effektiv och konsistent, erhåller man olika varianter av SLAM-lösningar. Vi kommer nedan att gå igenom några av de vanligaste och viktigaste varianterna.

2.5.1 EKF-SLAM

I den vanligaste varianten, EKF-SLAM, väljer man en state-space representation med ett antagande om additivt Gaussiskt vitt brus i kombination med ett Extended Kalman Filter (EKF). Filtret hanterar eventuellt olinjära mät- eller processmodeller genom sedvanlig linjärisering m.h.a. Taylor-utveckling kring det aktuella tillståndet.

EKF-SLAM är förmodligen en av de första SLAM-algoritmerna och är numera viktig mest av historiska skäl. EKF-SLAM lider av två stora nackdelar; uppdateringstiden är $\sim O(M^2)$ där M är antalet landmärken vilket gör att den bara kan hantera små omgivningar. Vidare är den ovanligt känslig för att rätt association görs mellan observationer och estimerade landmärken. Associationsproblemet är ett allmänt svårt problem som vi skall diskutera mer nedan.

2.5.2 FastSLAM

En annan viktig variant är FastSLAM [63] som har sin grund i sekventiella Monte Carlo metoder, eller *partikelfilter* som de också kallas. FastSLAM klarar därför direkt av icke-linjära processmodeller och icke-Gaussiska fördelningar av robotens tillstånd (pose). Varje partikel representerar en unik tillståndshistoria för roboten. *Givet* en viss historia (partikel) blir varje landmärke oberoende och kan därför representeras med en mängd av *oberoende* Gaussiska fördelningar. Till varje partikel hör en karta representerad av en sådan mängd. Detta innebär att uppdateringssteget blir mycket snabbare då endast de aktuella landmärkena behöver uppdateras. Detta sätt att partitionera tillståndsvektorn kallas *Rao-Blackwellisation*.

En annan stor fördel med att använda representationen med partiklar är att man kan använda olika dataassociationer per partikel [64]. Vanligtvis använder man en

dataassociations-hypotes för att uppdatera hela filtret vilket därför kräver extra stor omsorg i valet av denna hypotes.

FastSLAMs stora nackdel är att den representerar korrelationer mellan landmärken endast implicit genom spridningen av partikelmängden. Har man för få partiklar för en viss omgivning kommer partiklarna så småningom att dela en gemensam historia, vilket kan ge problem om man t.ex. kommer tillbaka till startpunkten efter en lång utfärd; den implicita beskrivningen av korrelationen mellan landmärkena nära startpunkten är då felaktig, vilket kan ge konvergensproblem om man jämför t.e.x med EKF-SLAM som beskriver korrelationerna explicit.

I praktiken är dock FastSLAM en av de bästa varianterna just nu och den skalar väldigt bra till stora omgivningar.

2.6 Forskningsfronten i SLAM

I denna del diskuteras⁴ olika metoder och ansatser för att hantera begränsningar i nuvarande SLAM-varianter.

2.6.1 Beräkningskomplexitet

Tids- och mätuppdateringsekvationerna (4),(5) är för naiva implementationer mycket beräkningskrävande och kan formuleras om för att bara beräkna det som verkligen behövs vid varje uppdatering.

- *Tillståndsmanipulering*: Eftersom processmodellen bara påverkar pose-estimatet behöver bara en del av tillstånden ingå i tidsuppdateringen.
- *Partitionerade Uppdateringar*: Vid en mätuppdatering med en ny observation så kommer en naiv implementering uppdatera estimaten för posen och *alla* landmärken i kartan. Genom att endast uppdatera de lokala landmärkena vid varje uppdatering och de globala landmärkena mer sällan, kan man vinna mycket i prestanda.
- *Glesning (Sparsification)*: En vanlig EKF-SLAM arbetar med tillståndsvektorer och kovariansmatriser för att representera Gaussiska fördelningar. Samma fördelningar kan istället beskrivas på informationsform (kanonisk form) där man använder informationsfiltret för tillståndsestimering. Fördelen är att för stora omgivningar kommer informationsmatrisen bli nästan gles, dvs. bara ett fåtal element i matrisen har stora värden. Genom att sätta de små värdena till noll erhåller man en gles matris som väldigt effektivt kan uppdateras [65,66].
- *Globala och relativa delkartor*: Den globala kartan delas upp i ett antal delkartor. Antingen delar delkartorna ett gemensamt (globalt) koordinatsystem eller så är de bara relaterade relativt varandra. Genom att använda SLAM fullt ut bara i en delkarta i taget så erhåller man stora prestandavinster. Det globala kartestimatet blir däremot konservativt (större osäkerheter än optimalt).

2.6.2 Dataassociation

Korrekt dataassociering, dvs. hur man parar ihop nya observationer av landmärken med de befintliga landmärkena i den estimerade kartan, är mycket viktigt för verkliga SLAM-

⁴ Avsnittet är baserat på [7] vilken rekommenderas för vidare läsning.

implementationer. Om någon associering blir fel, kan kartestimatet förvridas så att positionsbestämningen blir fel, vilket i sin tur kan göra att estimaten divergerar. En robust SLAM-metod måste kunna klara och återhämta sig från felaktiga associationer.

- *Batch Validering*: Många associationshypoteser över alla landmärken övervägs samtidigt. Tidiga SLAM-implementationer testade observationsassociationer ett landmärke i taget, vilket kan vara väldigt opålitligt om robotposen är osäker. Genom att studera kombinationen av många associationer samtidigt förbättrar man den totala tillförlitligheten. Två varianter av batch validering är *Joint Compatibility Branch and Bound (JCBB)* [67,68] som är en trädsökningsmetod, samt *Combined Constraint Data Association (CCDA)* [69] som är en grafsökningsmetod. Det finns varianter av båda metoderna som kan ge tillförlitliga associationer utan vetskap om robotens tillstånd över huvud taget [68,69]. Ofta räcker batch validering för att ge tillförlitlig dataassociation; om valideringen är tillräckligt restriktiv kommer även en felaktig association hänvisa till ett mycket närliggande landmärke vilket bara ger mindre störningar av konvergensen. I stora, komplexa miljöer kan det trots allt krävas mer noggranna metoder som t.ex. multihypotes följning [70] nedan.
- *Signaturvalidering*: Vissa typer av sensorer, t.ex. kameror, ger rika beskrivningar av omgivningen såsom färg, form och textur. Man kan knyta sådana signaturer (eller attribut) till landmärkena och använda dessa tillsammans med de vanliga geometriska egenskaperna för att förbättra dataassociationen [71,72].
- *Multihypotesgenerering*: Detta är en av de noggrannaste dataassociationsmetoderna eftersom man helt enkelt genererar nya spår för varje möjlig association. Eftersom detta leder till en kombinatorisk explosion av nya spår tar man hela tiden bort de mest osannolika spåren så att man bara bibehåller så många spår som man har beräkningsresurser till. Multi Hypothesis Tracking (MHT) har använts länge för följning av objekt i klotterrika miljöer [70] men har på grund av beräkningskraven (en karta per hypotes...) inte använts för SLAM. Man kan dock se FastSLAM som en multihypotes variant eftersom man där är fri att fritt välja dataassociation per partikel [64].

2.6.3 Omgivningsrepresentation

I början när man började arbeta med SLAM antog man att världen kunde modelleras som en mängd diskreta, enkla landmärken såsom punkter, linjer och cirklar. I mer realistiska omgivningar, och applikationer, räcker inte detta antagande utan mer rika beskrivningar av omgivningen måste kunna användas.

- *Senarelagd Fusion (Delayed Fusion)*: Brister hos sensorerna bidrar också till begränsningar i omvärldsmodellen. T.ex så kan en kamera ge bra bäringsinformation men inte avstånd. En sonar däremot ger bra avståndsinformation men sällan bra bäringsinformation. Om man direkt använder en sådan begränsad mätning genererar det en icke-Gaussisk fördelning över landmärkets position. Detta går bra om man använder mer generella sannolikhetsfördelningar i sin tillståndsestimering, såsom t.ex. en mix av Gaussiska fördelningar. Om man däremot vill använda Gaussiska positionsestimat för sina landmärken måste man använda s.k. senarelagd fusion (eng. *Delayed Fusion*) där man lagrar ett antal mätningar i en buffer tills man har full observerbarhet av landmärkena. Senarelagd fusion möjliggör också bättre tillståndsestimering m.h.a. smoothing, samt batch validering vilket ger säkrare dataassociationer.

- *Ickegeometrisk landmärken*: Genom att använda en modell av landmärkena som tillåter transformationer och registrering, t.ex. ett rutnät eller en OpenGL-modell, kan mycket mer generella beskrivningar än t.ex. punkter och linjer användas. När en robot observerar ett landmärke, görs en transformation av den aktuella landmärkesmodellen enligt de predikterade estimaten och resultatet kan jämföras med mätningen m.h.a. registreringstekniker.
- *3D-SLAM*: Traditionellt har SLAM arbetat med tvådimensionella topologiska kartor och där en robot har rört sig i ett plan. Men all teori generaliserar enkelt till fler dimensioner och t.ex. Davison et al. [73] har i ett mycket imponerande system, kallat MonoSLAM, med en enkel webbkamera visat SLAM i realtid (30 Hz). MonoSLAM tillåter kameran full rörelsefrihet med sex frihetsgrader (linjär rörelse samt rotationer i tre dimensioner).
- *Trajektoriebaserad SLAM*: I standardformuleringen av SLAM, se ekvation (1), söker vi den simultana fördelningen över robotens pose och landmärkenas position. I en alternativ formulering av SLAM estimerar man istället robotens hela historiska trajektorier genom omgivningen. Detta kan vara fördelaktigt om det är svårt att hitta diskreta, vällokaliserade landmärken och det är lättare att registrera och jämföra sensordata istället. Denna formulering har ingen explicit karta utan varje pose har istället en associerad mätvolym. Detta brukar kallas vybaserad objektigenkänning inom datorseendefältet. Genom att registrera de olika mätvolymerna mot varandra kan en karta genereras. FastSLAM är ett exempel på en variant av trajektoriebaserad SLAM eftersom varje partikel representerar en viss historisk trajektorier. Det finns många FastSLAM hybrider som använder sig av registrerade mätvolymmer istället för den traditionella landmärkesarten.[7] Andra varianter av trajektoriebaserad SLAM är t.ex. *Consistent Pose Estimation* (CPE) [71] och glesa versioner av ovanstående [7].
- *Inbyggd Tilläggsinformation*: Ibland vill man lagra data som är spatiellt lokaliserad, t.ex. lokal temperatur, fuktighet eller ljusförhållanden. Den trajektoriebaserade SLAM-varianten fungerar utmärkt för detta men den traditionella SLAM-versionen klarar inte detta lika bra. Nieto et al. [74] har utvecklat DenseSLAM som tillåter sådan spatiell lagring med hjälp av landmärkesestimater.
- *Dynamiska omgivningar*: Den kanske största begränsningen hos den vanliga SLAM-formuleringen är att den antar att omvärlden, och landmärkena, är statiska och att inget annat rör sig än roboten själv. (Se definitionen i avsnitt 2.3). Men en verklig värld är allt annat än statisk, speciellt i den applikation som denna rapport behandlar; människor, bilar, stolar och annat rör sig, momentant eller över tiden. Ofta är SLAM-varianter mycket redundanta då de följer fler landmärken än vad som är nödvändigt. Detta kan utnyttjas för att detektera rörliga objekt och antingen filtrera bort dem [75], eller utnyttja dem som en speciell typ av rörliga landmärken. I vår applikation är det senare att föredra då vi vill kunna följa egna och andras styrkor i omgivningen. En möjlig lösning är att kombinera SLAM med en multisensor-multimål tracker som kan följa rörliga mål. [76]

2.6.4 Multirobotsystem

I alla SLAM-varianter vi hittills har diskuterat har en ensam robot utforskat sin omgivning. I vår applikation vill vi kunna ha flera soldater som gemensamt bygger upp och delar en karta. Om vi antar att deras relativa positioner från början är okända är detta ett svårare

problem eftersom deras individuella kartor kan vara överlappande, förskjutna och roterade på okänt sätt relativt varandra.

Om vi däremot antar att deras relativa positioner är kända vid start, t.ex. genom GPS-följning fram till inträdet i byggnaden, kollapsar problemet till ett enklare problem där bidragen från de enskilda robotarna (soldaterna) kan behandlas som om de kom från en enskild robot.[8]

I allmänhet måste vi dock anta det svårare fallet av robusthetsskäl. Detta är en olöst fråga men några olika förslag på lösningar finns [77,78]. En del generaliserar trajektoriebaserad SLAM (där FastSLAM ingår) till multirobot-scenariot, men andra hävdar att förlusten av korrelationsinformation mellan landmärkena i dessa metoder förvärras ännu mer vid okända relativa positioner, och att andra lösningar krävs [78]. Speciellt intressant är arbetet av Freitas et al. [78], där en delad karta tas fram. Givet denna karta är de individuella robotarnas positioner oberoende och varje robot kan ha en egen uppsättning hypotespartiklar. Kartan betraktas som en uppsättning parametrar där inlärningsmetoder kan användas för att optimera parametrarna. Denna metod verkar inte dras med samma förlust av korrelationsinformation som t.ex. FastSLAM gör, och den verkar vara robust mot eventuella fel som görs då partiella kartor skall slås ihop.

2.7 Exempel på implementerade system

2.7.1 MonoSLAM

En väldigt intressant implementation av SLAM är *MonoSLAM* [73, 79] som enbart använder sig av en enkel handhållen webbkamera. Med denna erhåller man SLAM-estimat i 30Hz på en vanlig bärbar dator. Systemet tillåter translationer och rotationer av kameran i full 3D (6 frihetsgrader). Lokaliseringsnoggrannheten är några centimeter. Det nuvarande systemet klarar dock bara av relativt små omgivningar (~ ett rum). Författaren Davison, som fokuserar på realtidsprestanda, planerar dock vidare utveckling till mer realistiska omgivningar. En annan intressant aspekt på MonoSLAM är att den lagrar landmärkena som små bildtexturer, korrekt positionerade och orienterade i 3D. Det nuvarande systemet klarar av att generera driftfria mosaiker över omgivningen [80] och författaren planerar att vidareutveckla systemet mot s.k. superupplösning, där bildupplösningen av omgivningen kan bli betydligt högre än hos den genererande sensorn.

Det finns även andra författare som arbetar med s.k. monokulär SLAM, t.ex. [81]

2.7.2 Newman et al., "Explore and Return"

I detta experiment använde sig Newman et al. av EKF-SLAM i en medelstor omgivning inomhus [82]. Först fjärrstyrdes roboten genom omgivningen av en mänsklig operatör. Operatören kunde dock bara se omgivningen via den 3d-karta som roboten själv genererade online. Efter ett tag fick roboten ordern att autonomt ta sig tillbaka till utgångspunkten. Den planerade då själv en rutt tillbaka med hjälp av sin karta och kunde framgångsrikt återvända fullt autonomt.

2.7.3 Guivant och Nebot, SLAM i stora omgivningar utomhus

Guivant och Nebot har framgångsrikt använt SLAM i en utomhusapplikation med stora omgivningar (~300 x 300 meter) [83]. De visade hur man kan integrera RKT-GPS med SLAM för att förbättra prestanda, och också hur man skall hantera realtidsaspekter, fordon med höga hastigheter, kuperad terräng och dynamiskt klotter. Deras sparade data från

deras försök i Victoria Park finns tillgängliga på Internet och har blivit en populär benchmark inom SLAM-fältet.

2.7.4 Consistent Pose Estimation

Consistent Pose Estimation [71,84] är en lovande annorlunda SLAM-teknik som kommer från s.k. topologisk kartering och ger bra resultat i stora omgivningar inomhus.

3 Diskussion och förslag till fortsatt arbete

I kapitel 1 utgick vi från ett scenario där soldater närmar sig ett okänt hus med syfte att kartlägga det och bekämpa eventuellt motstånd. Med hjälp av scenariot satte vi upp en mängd krav ett tekniskt system måste uppfylla för att vara intressant i praktiken. Kapitel 2 introducerade SLAM-metoden och presenterade en rad varianter med olika för- och nackdelar. I detta kapitel kommer vi att diskutera hur väl de nuvarande SLAM-varianterna uppfyller de tekniska kraven vi ställde upp; vad finns, vad fattas och vad är på gång? Rapporten avslutas med ett förslag till fortsatt arbete.

3.1 Uppfyllelse av tekniska krav

Kraven som diskuteras nedan är de som presenterades i avsnitt 1.2

- *Sensorbörda & Realtidskrav*: MonoSLAM [79] som presenterades i avsnitt 2.7.1 visade att mycket kan göras i realtid med en mycket enkel sensor och ordinär beräkningshårdvara.
- *Distribuerad funktionalitet*: Att dela en gemensam karta på ett robust sätt är, som vi diskuterade i avsnitt 2.6.4, en av de största forskningsutmaningarna och där återstår mycket att göra. Det finns dock flera implementationer som har distribuerad funktionalitet [78] men dessa är begränsade i andra avseenden såsom t.ex. beräkningskrav och robusthet.
- *Dynamiska omgivningar*: I avsnitt 2.6.3 konstaterade vi att nästan alla SLAM-varianter förutsätter en statisk omgivning. Genom att kombinera SLAM med en multimål-multisensor tracker skulle man dock kunna filtrera bort de rörliga målen innan man skickar observationerna till SLAM [76]. På detta sätt erhåller man också automatisk följning av rörliga mål i omgivningen vilket i sig är ett viktigt mål i Red & Blue Force Tracking. Nackdelen med detta är ett ökat beräkningsbehov.
- *Resursbudget*: I avsnitt 2.6.1 diskuterades en rad åtgärder som kan vidtas för att minimera beräkningsbehoven, t.ex. partitionering och användning av delkartor. Genom att använda tekniken med delkartor kan man samtidigt minimera kommunikationsbehovet då endast små överlappande kartor behöver synkroniseras.
- *Kartkvalitet*: Traditionellt genererar SLAM ganska enkla tvådimensionella översiktskartor. Detta passar utmärkt för snabb kommunikation och för taktisk navigering då inga onödiga detaljer skall distrahera soldaten. Vi har dock sett att man t.ex. kan göra 3d-kartor och driftfri mosaiking med MonoSLAM om mer detaljerade kartor önskas.
- *A priori information*: Vi har diskuterat sammanslagning av kartor i multirobotsystem (avsnitt 2.6.4) och samma tekniker kan användas för att utnyttja extern förhandsinformation om denna är representerad på lämpligt sätt.
- *Okänd initialposition*: Om en ensam soldat har en okänd position är det inget problem eftersom alla SLAM-metoder kan hantera detta. Om flera samarbetande soldater har okända relativa positioner blir det mer besvärligt enligt multirobot-diskussionen i avsnitt 2.6.4. Olika ansatser till lösningar finns dock [8, 78]

3.1.1 Sammanfattning

Sammantaget kan vi konstatera att det finns SLAM-varianter som uppfyller kraven var och en för sig men att ingen verkar uppfylla alla. De krav som figurerar minst i litteraturen är *Dynamiska omgivningar* och *Distribuerad funktionalitet* men det verkar som dessa står på tur då flera artiklar har utkommit nyligen [76, 77, 78].

3.2 Förslag till fortsatt arbete

SLAM har en oerhörd potential att förbättra beslutsstöd och situationsförståelse för den enskilda soldaten eller stridsgruppen i situationer där externt stöd saknas eller är bristfällig. Ett prototypsystem som har potential att uppfylla de uppställda kraven presenteras nedan. Det baserar sig på litteraturöversiktarna i kapitel 2 och kan användas som utgångspunkt för fortsatt arbete.

Förutom att följa vidareutvecklingen av de tekniker som presenterats här rekommenderas forskning där syftet är att ta fram en ny SLAM-metod som uppfyller de tekniska krav som ställdes i kapitel 1, t.ex. baserat på nedanstående prototypsystem. Vi har sett att det finns SLAM-varianter som uppfyller kraven var och en för sig men att ingen uppfyller alla. Denna forskning kan förslagsvis göras som en fortsättning i projektet Sensorsystem för Urban Miljö (SUM) där denna förstudie är framtagen. Fokus för sådan forskning skulle vara realtidsprestanda ”i princip” med billig hårdvara, enligt inriktningen och andan av MonoSLAM[73, 79]; dvs. algoritmerna som används skall kunna skala till realtids-tillämpningar även om eventuella experiment- och simuleringsimplementationer inte gör det.

3.2.1 Prototypsystem

- *Sensorer*: Två miniaturiserade vidvinkelkameror sätts fast i soldatens hjälm tillsammans med miniaturiserade (MEMS⁵) accelerometrar och gyron. De två kamerorna används i stereo och ger därmed både bärings- och avståndsinformation om landmärkena till SLAM-algoritmen. Accelerometrarna hjälper SLAM-algoritmen vid kast, slag och snabba rotationer. Vidvinkeltegenskapen hos kamerorna hjälper här också till, eftersom landmärken i scenen stannar kvar i bilden även vid relativt stora förändringar av soldatens position och orientering. Kamerorna är passiva sensorer vilket hjälper soldaten att undvika upptäckt för fientliga sensorer. En annan fördel med att använda kameror framför andra sensorer är att video kan behövas i alla fall för andra behov, t.ex. för debriefing efter uppdraget. Kameratekniken är dessutom beprövad och väldigt kostnadseffektiv. Andra sensorer kan också utnyttjas av SLAM i mån av tillgänglighet, t.ex. ultraljud, radar, IR eller laser. Det är egentligen bara soldatens bekvämlighet, ekonomisk budget och tillgängliga batteri- och beräkningsresurser som sätter begränsningarna.
- *Kommunikation*: För att gemensamt kunna dela karta och information om rörliga mål, måste soldaterna kunna kommunicera data med varandra. Genom att använda tekniken med delkartor, beskrivet i sektion 2.6.1, kan man minimera kommunikationsbehovet då endast små överlappande kartor behöver synkroniseras. Vidare behöver bara kartsegment som ligger relativt nära en soldat kommuniceras till denne med hög

⁵ MicroElectroMechanical Systems (MEMS)

upplösning. Delkartor på större avstånd antas inte vara av lika stor vikt och kan därför kommuniceras med en lägre upplösning. Om soldaternas relativa positioner inte är kända måste SLAM-algoritmerna jämföra de individuella kartorna i försöken att hitta gemensamma delkartor, vilket är beräkningskrävande. Om kommunikationsutrustningen kan erbjuda relativ bärings- och avståndsinformation till de olika gruppmedlemmarna skulle detta underlätta kartdelningen betydligt. IFF-funktionalitet (Identification Friend or Foe) skulle då kunna integreras i den delade kartan på ett naturligt sätt.

- *SLAM- och följningsalgoritmer*: Vi låter en målföljare filtrera sensordetektionerna så att endast detektioner från stationära objekt skickas vidare till SLAM-algoritmen, som vi antar vara baserad på [78]. På detta sätt kan vi hantera dynamiska omgivningar, effektivt dela kartor mellan soldaterna och samtidigt detektera och följa rörliga objekt i omgivningarna. Eventuellt kan lokaliseringsdelen av SLAM vara baserad på någon MonoSLAM-variant för att säkerställa realtidskraven, medan karteringsdelen av SLAM får komma i andra hand i mån av beräkningsresurser, se t.ex. [85]. För att inte kräva att bägge kamerorna fungerar samtidigt, bör någon form av senarelagd datafusion användas (avsnitt 2.6.3). Detta gör att systemet kan extrahera djupinformation även om bara en kamera fungerar. Ett alternativ är att, i mån av beräkningsresurser, använda mer avancerade sannolikhetsfördelningar i landmärkes-estimeringen.

3.2.2 Forskningsverktyg

SLAM är ett exempel på ett system med signal- och bildbehandlingsalgoritmer. För att effektivt kunna utveckla, testa och verifiera sådana system krävs det att man kan göra bra simuleringar. Därför rekommenderas att ett framtida forskningsprojekt utnyttjar MSS-Lab⁶ (MultiSensorSimuleringsLaboratorium) för att göra tillförlitliga system- och sensorsimuleringar i relevanta och dynamiska syntetiska 3d-miljöer. För enklare sensor- och scenariosimuleringar, t.ex. under utvecklingsfasen, kan Microsoft Robotics Studio [86] eller Webots [87] också vara intressanta alternativ.

Det finns öppen källkod för många av de presenterade SLAM-varianterna, se t.ex. [88] samt översikten i [6] för vidare länkar. Öppna datamängder som kan användas för jämförelse och verifiering finns också i [6].

⁶ MSS-Lab finns på Institutionen för Sensorinformatik, FOI, Linköping.

4 Referenser

- [1] MARKUS - Markstridsutrustad soldat, Sverige, <http://www.mss.mil.se/article.php?lang=E&id=15184>
- [2] FIST - Future Infantry Soldier Technology, United Kingdom, <http://www.army-technology.com/projects/fist/>
- [3] IdZ (Infanterist der Zukunft) - Infantryman of the Future, Germany, <http://www.army-technology.com/projects/idz/>
- [4] Land Warrior, USA, <http://www.gdc4s.com/content/detail.cfm?item=aa0d1b86-ac8d-47ed-b59d-f8c2157beb7e&page=2>
- [5] ELSA, Enhanced Situational Awareness, http://www.defense-update.com/products/f/fist_sa.htm
- [6] Durrant-whyte, H.; Bailey, T. (2006). "[Simultaneous Localisation and Mapping \(SLAM\): Part I The Essential Algorithms](#)". *Robotics and Automation Magazine* **13**: 99-110.
- [7] Durrant-whyte, H.; Bailey, T. (2006). "[Simultaneous Localisation and Mapping \(SLAM\): Part II State of the Art](#)". *Robotics and Automation Magazine*, September 2006
- [8] Sebastian Thrun, Wolfram Burgard, Dieter Fox, Probabilistic Robotics, The MIT Press, September, 2005, ISBN-13: 978-0262201629
- [9] Michael Montemerlo, Sebastian Thrun, FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics, Springer 2007, ISBN-13: 978-3540463993
- [10] R. Smith, M. Self and P. Cheeseman, Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfon, editors, *Autonomous Robot Vehicles*, pages 167-193. Springer-Verlag, 1990.
- [11] H. Durrant-Whyte, D. Rye, and E. Nebot. Localisation of automatic guided vehicles. In G. Giralt and G. Hirzinger, editors, *Robotics Research: The 7th International Symposium (ISRR'95)*, pages 613-625. Springer Verlag, 1996.
- [12] Dave Cole, Paul Newman, Natural Feature Detection for 3D SLAM, <http://www.robots.ox.ac.uk/%7Epnewman/>
- [13] C.-Y. Chong, S. Mori, K.-C. Chang, and W.H. Barker, Architectures and Algorithms for Track Association and Fusion, *IEEE Aerospace and Electronic Systems*, vol. 15, no. 1, pp. 5-13, Jan. 2000.
- [14] E. Waltz and J. Llinas, Multisensor Data Fusion. Artech House, 1990.
- [15] D.L. Hall and J. Llinas, An Introduction to Multisensor Fusion, *Proc. IEEE*, vol. 85, no. 1, Jan. 1997.
- [16] Duncan Smith and Samseer Singh, Approaches to Multisensor Data Fusion in Target Tracking: A Survey, *IEEE Transactions On Knowledge And Data Engineering*, vol. 18., no. 12, December 2006.
- [17] C.-Y. Chong, S. Mori, K.-C. Chang, and W.H. Barker, Architectures and Algorithms for Track Association and Fusion, *IEEE Aerospace and Electronic Systems*, vol. 15, no. 1, pp. 5-13, Jan. 2000.

-
- [18] E. Waltz and J. Llinas, *Multisensor Data Fusion*. Artech House, 1990.
- [19] D.L. Hall and J. Llinas, An Introduction to Multisensor Fusion, *Proc. IEEE*, vol. 85, no. 1, Jan. 1997.
- [20] Duncan Smith and Samseer Singh, Approaches to Multisensor Data Fusion in Target Tracking: A Survey, *IEEE Transactions On Knowledge And Data Engineering*, vol. 18., no. 12, December 2006.
- [21] US Dept. of Defense, Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, Data Fusion Lexicon, 1991.
- [22] K. Wallenius, Support for Situation Awareness in Command and Control, *Proc. Seventh Int'l Conf. Information Fusion*, pp. 1117-1124, 2004.
- [23] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*. Artech House, 1992.
- [24] S.S. Blackman, Association and Fusion of Multiple Sensor Data, *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., pp. 187-218, Mass.: Artech House, 1990.
- [25] Y. Bar-Shalom, Extension of the Probabilistic Data Association Filter to Multitarget Tracking, *Proc. Fifth Symp. Nonlinear Estimation*, pp. 16-21, Sept. 1974.
- [26] R.J. Fitzgerald, Development of Practical PDA Logic for Multitarget Tracking by Microprocessor, *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., 1989.
- [27] E.A. Bloem and H.A. P. Blom, Joint Probabilistic Data Association Methods Avoiding Track Coalescence, *Proc. 34th IEEE Conf. Decision and Control*, vol. 3, pp. 2752-2757, 1995.
- [28] H.A.P. Blom and E.A. Bloem, Probabilistic Data Association Avoiding Track Coalescence, *IEEE Trans. Automatic Control*, vol. 45, no. 2, Feb. 2000.
- [29] S.D. O'Neil and L.Y. Pao, Multisensor Fusion Algorithms for Tracking, *Proc. 1993 Am. Control Conf.*, 1993.
- [30] S.S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston: Artech House, 1999.
- [31] K.R. Pattipatti, S. Deb, Y. Bar-Shalom, and R.B. Washburn, A New Relaxation Algorithm and Passive Sensor Data Association, *IEEE Trans. Automatic Control*, vol. 37, pp. 198-213, Feb. 1992.
- [32] Weisstein, Eric W, "NP-Problem." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/NP-Problem.html>
- [33] M. Winter and G. Favier, A Neural Network for Data Association, *Proc. 1999 IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1041-1044, Mar. 1999.
- [34] A.M. Aziz, M. Tummala, and R. Cristi, Fuzzy Logic Data Correlation Approach in Multisensor Multitarget Tracking Systems, *Signal Processing* 76, pp. 195-209, Elsevier, 1999.
- [35] H. Hong, H. Chong-zhao, Z. Hong-Yan, and W. Rong, Multi-Target Tracking Based on Multi-Sensor Information Fusion with Fuzzy Inference, *Control and Decision*, vol. 19, no. 3, pp. 272-276, 2004.
- [36] Y.M. Chen and H.C. Huang, Fuzzy Logic Approach to Multisensor Data Association, *Math. And Computers in Simulation* 52, pp. 399-412, Elsevier, 2000.

-
- [37] R. Kalman, A New Approach to Linear Filtering and Prediction Problems, *Trans. ASME, J. Basic Eng.*, no. 82, pp. 34-45, 1960.
- [38] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles Techniques and Software*. Artech House, 1993.
- [39] R. van der Merwe, E.A. Wan, and S. Julier, Sigma-Point Kalman Filters Nonlinear Estimation and Sensor Fusion—Applications in Integrated Navigation, *Proc. AIAA Guidance Navigation and Controls Conf.*, Mar. 2004.
- [40] S.J. Julier and J.K. Uhlmann, A New Extension of the Kalman Filter to Nonlinear Systems, *Proc. AeroSense: The 11th Int'l Symp. Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, 1997.
- [41] G.A. Ackerson and K.S. Fu, On State Estimation in Switching Environments, *IEEE Trans. Automatic Control*, vol. 5, no. 1, 1970.
- [42] J.M. Bruckner, H.R.W. Scott, and G.R. Rea, Analysis of Multimodal Systems, *IEEE Trans. Aerospace and Electronic Systems*, vol. 9, no. 6, pp. 883-888, 1973.
- [43] Y. Bar-Shalom and H.A.P. Blom, The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients, *IEEE Trans. Automatic Control*, vol. 33, no. 8, Aug. 1988.
- [44] L. Hong, Optimal Multiresolutional Distributed Filtering, *Proc. 31st Conf. Decision and Control*, Dec. 1992.
- [45] L. Hong, Multiresolutional Distributed Filtering, *IEEE Trans. Automatic Control*, vol. 39, no. 4, Apr. 1994.
- [46] L. Hong, J.R. Werthmann, G.S. Bierman, and R.A. Wood, A Multiresolutional Approach to Target Tracking, *Proc. IEEE 1993 Nat'l Aerospace and Electronics Conf.*, vol. 1, pp. 388-392, 1993.
- [47] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, Novel Approach to Non-Linear/Non-Gaussian Bayesian State Estimation, *IEE Proc.-F*, vol. 140, no. 2, Apr. 1993.
- [48] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Trans. Signal Processing*, vol. 50, no. 2, Feb. 2002.
- [49] A.M. Flynn, Combining Sonar and Infrared Sensors for Mobile Robot Navigation, *Int'l J. Robotics Research*, pp. 5-14, Dec. 1988.
- [50] H.S. Abdel-Aty-Zohdy and R.L. Ewing, Intelligent Information Processing Using Neural Networks and Genetic Algorithms, *Proc. 43rd Midwest Symp. Circuits and Systems*, pp. 840-845, Aug. 2000.
- [51] Cortes, C., Vapnik, V., Support Vector Networks. *Machine Learning* 20, 273-297, 1995.
- [52] P.J. Escamilla-Ambrosio and N. Mort, Multisensor Data Fusion Architecture Based on Adaptive Kalman Filters and Fuzzy Logic Performance Assessment, *Proc. Fifth Int'l Conf. Information Fusion*, vol. 2, pp. 1542-1549, 2002.
- [53] J.Z. Sasiadek and P. Hartana, Sensor Data Fusion Using Kalman Filter, *Proc. Third Int'l Conf. Information Fusion*, vol. 2, pp. WED5/19-WED5/25, July 2000.
- [54] A.P. Dempster, A Generalisation of Bayesian Inference, *J. Royal Statistical Soc.*, vol. 30, pp. 205-247, 1968.

-
- [55] G. Shafer, A Mathematical Theory of Evidence. *Princeton Univ. Press*, 1976.
- [56] S. Challa and D. Koks, Bayesian and Dempster-Shafer Fusion, *Sadhana*, vol. 29, part 2, pp. 145-174, Apr. 2004.
- [57] S.P. Chaudhuri and S. Das, Neural Networks for Data Fusion, *Proc. IEEE Int'l Conf. Systems Eng.*, 1990.
- [58] S.A. Storm, An Investigation into the Effects of Correlation in Sensor Fusion, thesis, Dept. Air Force, Air Univ., US Air Force Inst. of Technology, Mar. 2003.
- [59] J. Kittler, Multi-Sensor Integration and Decision Level Fusion, *Proc. DERA/IEEE Workshop Intelligent Sensor Processing (Ref. No. 2001/050)*, pp. 6/1-6/6, 2001.
- [60] J. Kittler and F.M. Alkoot, Sum Versus Vote Fusion in Multiple Classifier Systems, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, Jan. 2003.
- [61] R.R. Brooks, P. Ramarathan, and A.M. Sayeed, Distributed Target Classification Tracking in Sensor Networks, *Proc. IEEE*, vol. 91, no. 8, Aug. 2003.
- [62] T. Clouqueur, P. Ramanathan, K.K. Saluja, and K.-C. Wang, Value-Fusion Versus Decision-Fusion for Fault-Tolerance in Collaborative Target Detection in Sensor Networks, *Proc. Fusion 2001 Conf.*, 2001.
- [63] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI National Conference on Artificial Intelligence*, pages 593-598, 2002.
- [64] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *IEEE International Conference on Robotics and Automation*, pages 1985-1991, 2003.
- [65] S. Thrun, D. Koller, Z. Ghahmarani, H. Durrant-Whyte, and A. Ng. Simultaneous localization and mapping with sparse extended information filters: Theory and initial results. Technical report, Carnegie Mellon University, 2002.
- [66] S. Thrun, Y. Liu, D. Koller, A. Ng, and H. Durrant-Whyte. Simultaneous localisation and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693-716, 2004.
- [67] J. Neira and J.D. Tard'os. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890-897, 2001.
- [68] J. Neira, J.D. Tard'os, and J.A. Castellanos. Linear time vehicle relocation in SLAM. In *IEEE International Conference on Robotics and Automation*, 2003.
- [69] T. Bailey. Mobile Robot Localisation and Mapping in Extensive Outdoor Environments. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2002.
- [70] Y. Bar-Shalom and T.E. Fortmann. Tracking and Data Association. Academic Press, 1988.
- [71] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318-325, 1999.
- [72] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *IEEE International Conference on Robotics and Automation*, 2006.

-
- [73] A.J. Davison, Y.G. Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.
- [74] J. Nieto, J. Guivant, and E. Nebot. The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM. In *IEEE International Conference on Robotics and Automation*, pages 391–396, 2004.
- [75] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 1557–1563, 2003.
- [76] C.C. Wang, C. Thorpe, and S. Thrun. On-line simultaneous localisation and mapping with detection and tracking of moving objects. In *IEEE International Conference on Robotics and Automation*, pages 2918–2924, 2003.
- [77] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, B. Stewart, Distributed Multi-Robot Exploration and Mapping
- [78] Ruben Martinez-Cantin, José A. Castellanos, Nando de Freitas, Multi-Robot Marginal-SLAM, 2007
- [79] 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 (PAMI)*, 29(6), pp 1052--1067, 2007
- [80] Javier Civera, Andrew J. Davison, J. A. Magallón and J. M. M. Montiel, Drift-Free Real-Time Sequential Mosaicing, *IJCV 2008* (accepted for publication),
- [81] Brian Williams, Georg Klein and Ian Reid, Real-Time SLAM Relocalisation, *Proc International Conference on Computer Vision, Rio de Janeiro*, Oct 2007
- [82] P.M. Newman, J.J. Leonard, J. Neira, and J. Tardos. Explore and return: Experimental validation of real time concurrent mapping and localization. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1802–1809, 2002.
- [83] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robotics and Automation*, 17(3):242-257, 2001.
- [84] K. Konolige. Large-scale map-making. In *National Conference on AI (AAAI)*, pages 457-463, 2004.
- [85] G. Klein and D.W. Murray, Parallel Tracking and Mapping for Small AR Workspaces, *Proc International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007
- [86] Microsoft Robotics, <http://msdn.microsoft.com/en-us/robotics/default.aspx>
- [87] Webots, <http://www.cyberbotics.com/>
- [88] Tim Bailey, Matlab-Simulators for SLAM, http://www-personal.acfr.usyd.edu.au/tbailey/software/slam_simulations.htm