

FREDRIK NÄSSTRÖM, FREDRIK BISSMARCK,  
VIKTOR DELESKOG, FREDRIK HEMSTRÖM,  
MAX HOLMBERG, JÖRGEN KARLHOLM,  
JONAS NORDLÖF, JONAS NYGÅRDS,  
KARL-GÖRAN STENBORG, NICLAS WADSTRÖMER



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.

Fredrik Näsström, Fredrik Bissmarck,  
Viktor Deleskog, Fredrik Hemström,  
Max Holmberg, Jörgen Karlholm,  
Jonas Nordlöf, Jonas Nygårds,  
Karl-Göran Stenborg, Niclas Wadströmer

# Intelligenta spaningsfunktioner 2016-2018

Slutrapport

Titel	Intelligenta spaningsfunktioner, slutrapport
Title	Intelligent reconnaissance functions, final report
Rapportnr/Report no	FOI-R-4648--SE
Månad/Month	December
Utgivningsår/Year	2018
Sidor/Pages	35 p
Kund/Customer	Försvarsmakten
Forskningsområde	Sensorer och signaturanpassning
FoT-område	Sensorer och signaturanpassning
Projektnr/Project no	E72701
Godkänd av/Approved by	Johan Allgurén
Ansvarig avdelning	Ledningssystem
Exportkontroll	Innehållet är granskat och omfattar ingen information som är underställd exportkontrollagstiftningen.

Detta verk är skyddat enligt lagen (1960:729) om upphovsrätt till litterära och konstnärliga verk, vilket bl.a. innebär att citering är tillåten i enlighet med vad som anges i 22 § i nämnd lag. För att använda verket på ett sätt som inte medges direkt av svensk lag krävs särskild överenskommelse.

This work is protected by the Swedish Act on Copyright in Literary and Artistic Works (1960:729). Citation is permitted in accordance with article 22 in said act. Any form of use that goes beyond what is permitted by Swedish copyright law, requires the written permission of FOI.

*Omslagets bildcollage: Översta raden från vänster till höger, den vänstra bilden visar spaning från stridsfordon 90 (Foto: Janne Bohman/Försvarsmakten) och den högra bilden visar en IR-bild där en automatisk måligenkänningsalgoritm korrekt har känt igen en människa. Nedersta raden från vänster till höger, den vänstra bilden visar pansarterrängbil 360 (Foto: Jan Bohman/Försvarsmakten) och den högra bilden visar en IR-bild där en automatisk måligenkänningsalgoritm korrekt har känt igen en lastbil och en personbil.*

## Sammanfattning

I denna rapport presenteras arbetet som utförts i projektet Intelligent spaningsfunktioner. I projektet har forskning bedrivits med syfte att göra framtida sensorsystem mer intelligenta. Genom att vara mer intelligenta kan framtida sensorsystem ge sensoroperatörer bättre situationsuppfattning, högre systemtilltro, mindre risk för felhandlingar (eng. human error) och lägre mental arbetsbelastning. Projektet har bedrivit forskning inom områdena: automatisk måligenkänning, positionering och sensorplanering.

Inom måligenkänning har två måligenkänningsalgoritmer utvecklats för detektion och igenkänning av olika objekt. De framtagna måligenkänningsalgoritmerna är bra på att känna igen både rörliga och stillastående objekt i komplexa miljöer. Algoritmerna är baserade på djupa neuronnät. I projektet har vi arbetat med civila objekt, men algoritmerna kan tränas att känna igen andra objekt såsom stridsfordon, fartyg och helikoptrar.

Inom automatisk positionering utan att använda satellitnavigering har två försök genomförts för att analysera positioneringsprestandan. I det första försöket har ett enskilt fordon färdats efter en bana varefter det totala positioneringsfelet mättes upp. I det andra försöket har ett mål detekterats med en IR-sensor och mätts in från två självpositionerande fordon. Målets position, som i detta fall var en person, har bestämts i ett för fordonen gemensamt koordinatsystem. Det första försöket visade att positioneringssystemet fungerar i olika terräng med en liten osäkerhet, och det andra försöket visade att den relativa osäkerheten blir mindre för två positioneringssystem som samverkar än osäkerheten för var och ett av positioneringssystemen. Inmätningosäkerheten för ett mål som trianguleras in beror i huvudsak på den relativa vinkelosäkerheten och eftersom den blir mindre med samverkande positioneringssystem så minskar inmätningosäkerheten av målet.

Inom spaning med samverkande styrbara sensorer har projektet visat hur dessa kan användas för att öka spaningstäckningen och förbättra målsparskvaliteten. Resultaten visar att sensorer som tillåts att styras automatiskt kan höja spaningsförmågan. Genom att de styrs automatiskt och data behandlas av algoritmer tillåts en hög svephastighet, vilket bidrar till en högre målsparprecision. När ett hot har detekterats av någon sensor väljs den sensor som ger mest ökning av målprecisionen och övriga sensorer kan fortsätta spana. Automatisk sensorplanering bidrar således till ett mer resurseffektivt användande av tillgängliga sensorer.

Nyckelord: måligenkänning, sensorsamverkan, fusion, sensorstyrning, positionering, maskininlärning

## Summary

This report presents the work carried out in the project Intelligent reconnaissance functions. In the project, research has been conducted with the aim to make future sensor systems more intelligent. By being more intelligent, future sensor systems provide sensor operators with better situational awareness, higher system confidence, reduced risk of human errors and low mental workload. The project has conducted research in the area of automatic target recognition, positioning without using satellite navigation, and sensor planning.

Within target recognition, two automatic target recognition algorithms have been developed for automatic detection and recognition of different objects. The developed target recognition algorithms are good at recognizing both moving and stationary objects in complex environments. The algorithms are based on deep neural networks. In the project we have worked with civilian objects, but the algorithms can learn to recognize other objects such as combat vehicles, ships and helicopters.

Within automatic positioning without using satellite navigation, two tests have been made to analyze the positioning performance. In the first test, an individual vehicle has traveled along a path, after which the total positioning error is measured. In the second test, a target has been detected with an IR sensor and is measured from two self-positioning vehicles. The position of the target, which in this case was a person, has been estimated in a common coordinate system for the vehicles. The first test showed that the positioning system has good performance in different terrains with a small uncertainty, and the second test has showed that the relative uncertainty becomes smaller for two positioning systems that cooperate than the uncertainty for each of the individual positioning systems. The measurement uncertainty of a triangulated target is mainly due to relative angular uncertainty, and it becomes smaller with cooperative positioning system so the measurement uncertainty for the target decreases.

Within reconnaissance with collaborating sensors, the project has demonstrated how these can be used to increase reconnaissance coverage and improve target track quality. The results show that sensors that are allowed to be controlled automatically can increase the reconnaissance performance. By automatically controlling and processing data at a high scan rate a higher track quality is achieved. Automatic sensor planning therefore gives a higher reconnaissance capability, because when a threat has been detected by any sensor, the sensor which maximizes target precision is selected and other sensors can continue to scout. This contributes to a more resource efficient use of available sensors.

Keywords: target recognition, collaboration, fusion, sensor control, positioning, machine learning

## Innehåll

<b>1</b>	<b>Inledning</b>	<b>7</b>
<b>2</b>	<b>Intelligenta spaningsfunktioner</b>	<b>8</b>
2.1	Måligenkänning .....	8
2.1.1	Datainsamling och annotering .....	8
2.1.2	Faltningsnät för detektion och igenkänning .....	9
2.1.3	Mask R-CNN .....	10
2.1.4	RetinaNet .....	11
2.2	Positionering .....	12
2.2.1	EKF-SLAM .....	12
2.2.2	Smoothing and Mapping - SAM .....	13
2.2.3	Beskrivning av LiDAR-baserat positioneringssystem .....	15
2.3	Spaning och sensorplanering .....	15
<b>3</b>	<b>Sensorsystemet MSP</b>	<b>18</b>
<b>4</b>	<b>Resultat för signalbehandlings-algoritmerna</b>	<b>20</b>
4.1	Operatörsstöd med måligenkänning .....	20
4.2	Analys av positioneringsprestanda .....	22
4.2.1	Kvalitativa resultat av egenpositionering .....	23
4.2.2	Jämförelse mellan GPS-stödd och GPS-fri positionering .....	25
4.2.3	Positionering av utpekade mål .....	26
4.2.4	Diskussion .....	29
4.3	Förbättrad spaningsförmåga med automatisk sensorplanering .....	30
<b>5</b>	<b>Fortsatt arbete</b>	<b>34</b>
<b>6</b>	<b>Referenser</b>	<b>35</b>

## Förkortningar

FOV	Field of view
GNSS	Global Navigation Satellite Systems
GT-SAM	Georgia Tech Smoothing and Mapping
IMU	Inertial Measurement Unit
ISAM	Incremental Smoothing And Mapping
LiDAR	Light Detection And Ranging
LWIR	Long-wavelength infrared
MSP	MultiSensorPlattform
MSSLab	MultiSensorSimuleringsLab
RPA	Remotely Piloted Aircraft – fjärrstyrt flygplan
RPAS	Remotely Piloted Aircraft System – fjärrstyrt flygsystem
SAM	Smoothing And Mapping
SLAM	Simultaneous Localization And Mapping
TLE	Target Location Error

# 1 Inledning

Syftet med projektet Intelligent spaningsfunktioner har varit att studera hur avancerad signalbehandling kan användas för att göra sensorsystem mer ”intelligenta”. Syftet är att underlätta sensoroperatörens arbete och därmed öka spaningens effektivitet och kvalitet. Målsättningen med de nya signalbehandlingsalgoritmerna som tagits fram i projektet är att de ska bidra till framtida operatörssystem som ska ge bättre situationsuppfattning, högre systemtilltro, mindre risk för felhandlingar (eng. human error) och låg mental arbetsbelastning för operatören.

Utvecklingen inom signal- och bildbehandlingsområdet går snabbt över hela världen. Detta gör att nya automatiska funktioner hela tiden utvecklas. Fyra fördelar med automatiska funktioner är att de vanligen leder till:

- Ökad effektivitet (precision/snabbhet)
- Ökad säkerhet
- Minskad arbetsbelastning (mental och fysisk)
- Förbättrad ekonomi

Även om nya automatiska funktioner utvecklas för signal- och bildbehandling behövs fortfarande en människa som utifrån det övergripande sammanhanget tolkar skeenden och prioriterar sensorresurser. Människan är därför ofta oersättlig, precis som i många andra komplexa system. Förr eller senare är det alltid en människa som ansvarar för att styra, justera och omkonfigurera systemet och dess automatiska funktioner beroende på aktuella krav. Systemen måste därför konstrueras så att människan kan använda dem på ett effektivt och säkert sätt.

En vanlig orsak till att operatörer kan ha svårt att hantera komplexa system på avsett sätt är bristen på återkoppling (feedback). Operatören får helt enkelt för lite eller otydlig information om systemets status och hur den utvecklas [1]. Om systemet och operatören inte kan kommunicera på ett tillfredsställande sätt kan operatören uppfatta att systemet gör helt oväntade saker, s.k. Automation Surprises. Det här var tidigare ett problem inom den civila luftfarten när man började använda automatiserade funktioner som resulterade i en helt ny typ av tillbud och olyckor [2]. Just bristen på återkoppling till operatören var en av orsakerna till de här problemen, vilket gjorde det mycket svårt för operatören att ingripa och göra rätt saker för att hantera störningar. En felinmatad siffra eller ett glömt kommatecken resulterade i att systemet stumt utförde vad det blivit instruerat att göra, hur fel detta än var och utan att visa piloterna resultatet [3].

Det som avgör hur bra ett operatörsgränssnitt fungerar är vilka förutsättningar det ger för att förstå farkostens förmåga att hantera förväntade situationer och att koordinera hur uppgiften ska lösas [4]. Hur omfattande koordinationsproblemet är beror på vilka krav situationen ställer och farkosternas förmåga att upprätthålla en acceptabel prestationsnivå utan operatörens medverkan. Om koordinationsproblemet är svårt så minskar antalet farkoster som operatören kan hantera. Det är därför viktigt med bra gränssnitt för svåra koordinationsproblem så att operatören inte blir överbelastad.



## 2 Intelligent spaningsfunktioner

Målet med arbetet i projektet har varit att ta fram generisk funktionalitet för samverkande sensorsystem för att öka effektiviteten och kvaliteten på spaningen, men samtidigt underlätta sensoroperatörens arbete. Forskningen i projektet har inriktats mot automatisk måligenkänning, positionering av mål i olika miljöer från rörliga sensorplattformar samt styrning av sensorer för optimal informationsinhämtning vid spaning mot markmål (människor och fordon).

I projektet har vi valt att fokusera på signalbehandling för markarenan, i synnerhet sensorer för stridsfordon, men algoritmerna som beskrivs i rapporten är generella och kan användas även i övervakningssystem, RPAS (Remotely Piloted Aircraft System, fjärrstyrt flygsystem) fartyg, helikoptrar, flygplan, robotar etc.

### 2.1 Måligenkänning

Målsättningen med måligenkänningsarbetet har varit att automatiskt kunna stötta vagnchef och skytt i upptäckt och identifiering av hot. Måligenkänningsarbetet kan även användas vid fast spaning för att skapa kvalificerade obemannade observationsplatser som automatiskt kan generera spaningsrapporter, vilket minskar kommunikationsbehovet och därmed risken för upptäckt.

Måligenkänningsarbetet har varit inriktat mot att analysera moderna metoder för detektering och igenkänning av objekt i video från främst termiska IR-sensorer. Under de senaste åren har mycket stora framsteg gjorts inom objektigenkänning i bilder. Det stora genombrottet kom 2012 då en forskargrupp vid University of Toronto publicerade ett arbete [5] där användning av djupa neuronät (eng. deep learning) uppnådde radikalt bättre klassificeringsprestanda än tidigare metoder. Detta har lett till ett paradigmskifte inom området och en närmaste explosiv utveckling, till stor del driven av de stora IT-jättarna (Google, Microsoft, Facebook, Amazon, Baidu, Tencent, m.fl.).

Ett djupt neuronät innehåller ett mycket stort antal (typiskt 10-1000 miljoner) parametrar vars värden bestäms under en träningsprocess. Under träningen anpassas parametrarna successivt för att minska nätverkets prediktionsfel på träningsdata. Ju fler parametrar nätverket innehåller, desto fler träningsexempel behövs för att nätverket ska kunna generalisera väl till nya data. Träningen är extremt beräkningskrävande och kräver oftast särskild hårdvara för att vara praktiskt genomförbar. Grafikprocessorer (GPU:er) har visat sig ha en arkitektur som väl lämpar sig för neuronätsberäkningar. Beräkningarna kan dessutom fördelas mellan flera GPU:er som parallellt bearbetar data, vilket ger en ytterligare uppsnabbning.

I projektet har vi av praktiska skäl valt att studera metoder för att detektera och särskilja mellan sju vanligt förekommande objekttyper: bilar, bussar, cyklister, fotgängare, lastbilar, motorcyklister, samt motorredskap (traktorer, entreprenadmaskiner, etc.).

#### 2.1.1 Datainsamling och annotering

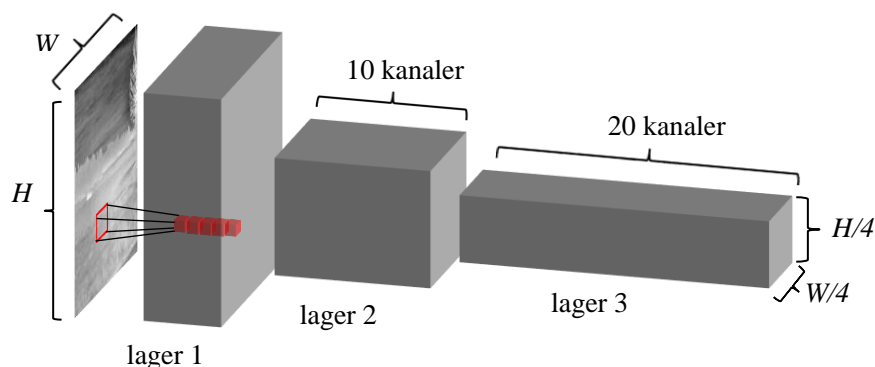
Bilder till träningen av de djupa neuronäten har samlats in med en fordonsmonterad ökyld termisk IR-sensor från olika miljöer i Linköping med omnejd under olika årstider och väder. Endast en delmängd av bilderna har annoterats (markerats per pixel) då detta arbete är mycket tidskrävande. Verktyg för att stötta och snabba upp processen bör därför prioriteras framöver. Ett exempel på annotering visas i figur 1. Färgerna markerar här olika objektklasser. Samtliga objekt ur de aktuella klasserna större än en minimistorlek markeras. Utöver de ovan nämnda sju klasserna används också en kategori "Ignorera". Denna används för regioner som inte ska klassificeras under träningsprocessen. Det gäller exempelvis objekt som är delvis skylda (rosa i bilden) eller stora grupper som bilparkeringar och folksamlingar.



Figur 1. Annotering av bilder på pixelnivå. Färgerna markerar olika objektklasser samt den särskilda kategorin "ignorera".

### 2.1.2 Faltningssnät för detektion och igenkänning

Med djupa neuronät menas inom bildanalys nästan alltid faltningssnät (eng. convolutional neural networks, CNN). I figur 2 visas schematiskt hur ett faltningssnät skapar en serie representationer av bilden med allt lägre spatiell upplösning men allt flera kanaler (värden per position). En sådan representation kallas ofta en särdragskarta (eng. feature map). Ett typiskt faltningssnät kan ha mellan 50 och 200 lager, där det sista lagret har kanske 1/16- eller 1/32-del av bildens upplösning, men i många fall flera tusen kanaler. Olika tillämpningar som objekt-detektion och semantisk segmentering kan använda samma arkitektur för särdrags-extraktion. Den aktuella tillämpningen adresseras genom att lägga till operationer (typiskt ytterligare faltningssnät) som verkar på ett eller flera lager i särdrags-extraktorn.

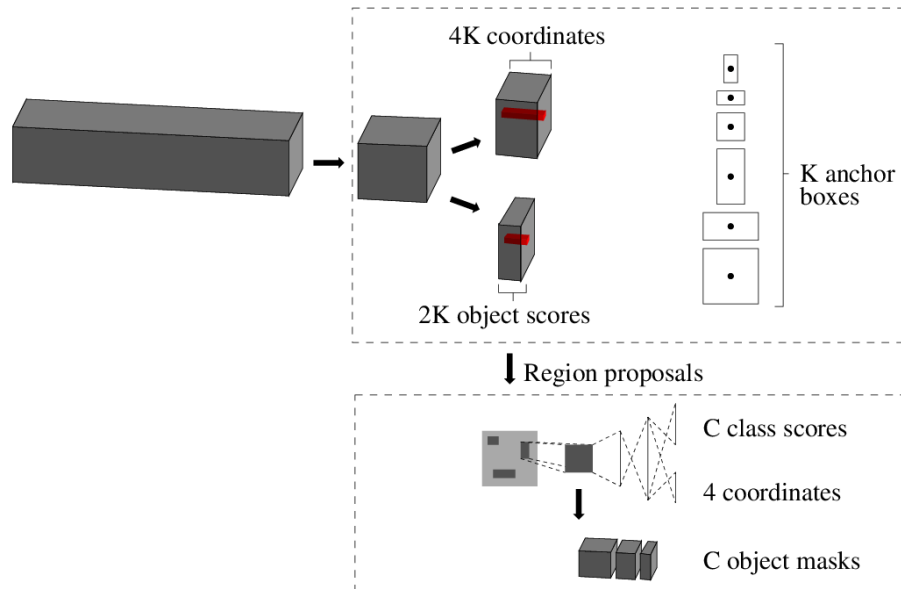


Figur 2. Särdrags-extraktion i ett faltningssnät. Varje lager i nätverket består av en uppsättning korrelationsfilter; här har det första (av tre lager) fem filter. Varje filter skapar, när det appliceras på data från föregående lager, en kanal i lagrets representation av bilden. Typiskt minskas successivt lagrens spatiella utbredning, samtidigt som antalet kanaler ökar. Kanalerna i varje punkt representerar en omgivning i bilden; i det första lagret en liten region (röd kvadrat i bilden) stor som filtrets utbredning, senare allt större områden.

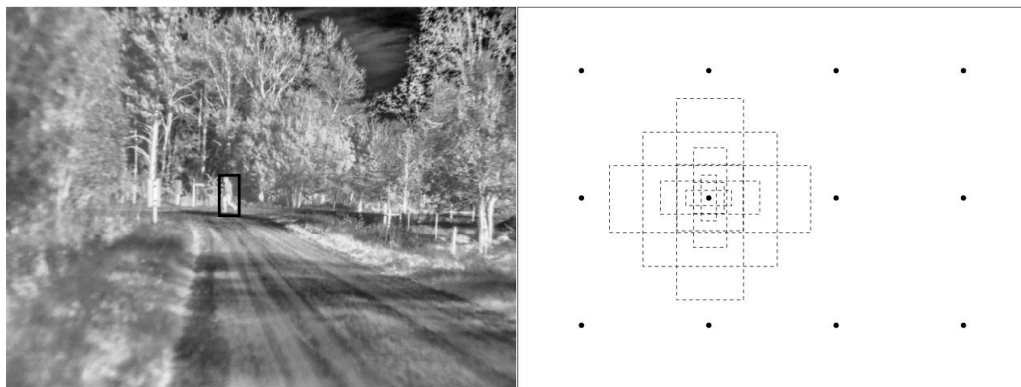
Metoder för faltningssnät-baserad detektion och igenkänning kan grovt indelas i två kategorier. I tvåstegsalgoritmer skannas bilden först för att identifiera objektlika regioner. Dessa extraheras sedan och klassificeras i ett andra steg. I enstegsalgoritmer sker såväl detektion som klassificering i ett enda steg. Historiskt har enstegsalgoritmer varit snabbare, men haft sämre klassificeringsprestanda än tvåstegsmetoder. Det senaste året har dock snabba enstegsmetoder introducerats som prestandamässigt kan mäta sig med tvåstegsalgoritmer.

### 2.1.3 Mask R-CNN

Faster R-CNN [6] är en tvåstegsalgoritm för objekt-detektion och klassificering vars prestanda visat sig mycket svåra att överträffa. He et al. [7] modifierade algoritmen så att den för varje objekt även genererar en mask som indikerar vilka bildpunkter som hör till objektet. Algoritmen, kallad Mask R-CNN<sup>1</sup>, beskrivs i figur 3 och figur 4.



Figur 3. Mask R-CNN. Data från det sista lagret i särdragsextraktorn (jmf figur 2) processas efter en dimensionsreduktion parallellt i två mindre faltningssnät. För varje spatiell position (rött) skattas för  $K$  ankarboxar (se figur 4) sannolikheten att de innehåller ett objekt, samt hur boxen ska justeras för att bättre omsluta objektet. Boxarna med störst målsannolikhet (eng. region proposals) går vidare till nästa steg, där motsvarande områden i särdragskartan extraheras och normaliseras i storlek och form. Därefter processas boxarna i två fullt kopplade neuronnät som skattar sannolikheter för olika objektclasser, respektive en ytterligare justering av boxen som omsluter objektet. Den extraherade regionen processas även i ett faltningssnät som genererar en mask per klass indikerande vilka bildpunkter som bedöms tillhöra objektet.

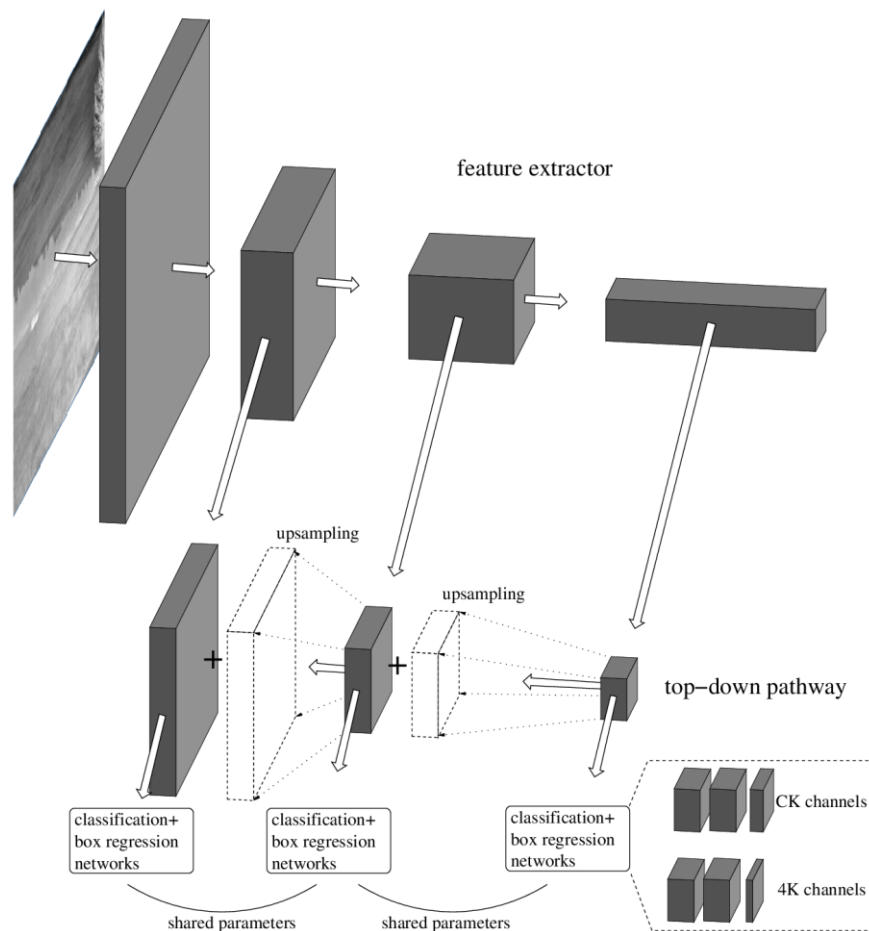


Figur 4. Vänster: Bild i upplösningen  $640 \times 480$  pixlar med ett annoterat objekt (person). Höger: Ankarboxar (schematiskt) för en position i en särdragskarta med upplösningen  $4 \times 3$  punkter. Boxarna refererar till regioner i bilden.

<sup>1</sup> Best Paper Award (Marr Prize) vid ICCV 2017.

## 2.1.4 RetinaNet

RetinaNet<sup>2</sup> [8], figur 5, är den första enstegsalgoritmen med detektions- och klassificeringsprestanda som kan konkurrera med Faster R-CNN. RetinaNet utnyttjar information från flera olika lager i särdragsextraktorn. En viktig innovation är att information tillåts propagera från de senare lagren tillbaka till de tidigare ("top-down"). Detta tillför kontext till beslutsunderlaget på de lägre nivåerna. Varje lager i top-downströmmen är kopplat till två faltningssätt, vilka för varje position skattar sannolikheter för olika objektklasser i ankarboxar (jmf figur 4), respektive hur boxarna ska justeras för att bättre omsluta objekten.



Figur 5. RetinaNet. Nätverket tappar av information från flera lager i en särdragsextraktor. Information propagerar sedan top-down från de senare särdragslagren tillbaka till de tidigare lagren. Informationen i de senare lagren har högre abstraktionsnivå och täcker större områden i bilden, vilket tillför en kontext till de tidigare lagren. Till varje top-downlager är sedan två faltningssätt kopplade; ett för skattning av sannolikheterna för  $C$  objektklasser i  $K$  ankarboxar per position (jmf figur 4), och ett för skattning av hur boxarnas position ska korrigeras för att bättre omsluta objekten (4 koordinater per box).

Kritiskt för att RetinaNet ska fungera är användningen av en kostnadsfunktion kallad *focal loss*. Den adresserar en viktig skillnad mellan en- och tvåstegsalgoritmer, nämligen behandlingen av asymmetrin mellan mängden objekt- och bakgrundsexempel i träningsdata. Normalt sett är det mycket vanligare att en ankarbox innehåller bakgrund än ett objekt. I en enstegsalgoritm kan detta medföra att kostnadsfunktionen helt domineras av bakgrund, och träningen fokuseras därmed på att korrekt klassificera dessa. Tvåstegsalgoritmer som Faster R-CNN har inte detta problem eftersom det första steget identifierar en mindre mängd (något tusental) regioner med högst sannolikhet att innehålla objekt, och dessa skickas sedan

<sup>2</sup> Best Student Paper Award vid ICCV 2017.

vidare för analys i det andra steget. Dessutom kan ibland en slumpmässig delmängd av dessa väljas ut (några hundra) för att bättre balansera andelarna objekt och bakgrund (typiskt i förhållande 1:3). Man kan säga att *focal loss* modifierar kostnadsfunktionen genom att (1) ge varje bakgrundsexempel lägre vikt, vilket förbättrar balansen, och (2) vikta ner exempel som är lätta att klassificera (de flesta bakgrundsexemplen), vilket fokuserar träningen på svåra exempel (objekt och klotter).

## 2.2 Positionering

Att med god noggrannhet veta position och orientering för sina sensorer är viktigt vid många militära operationer som till exempel vid inmätning av mål för indirekt bekämpning eller för att kunna skapa en gemensam lägesbild mellan flera sensorplattformar. Vid en konflikt mot en kvalificerad motståndare är det mycket sannolikt att olika telekrigssystem kommer att användas för störning och vilseledning. Därmed kommer satellitnavigering inte vara ett tillförlitligt hjälpmedel för positionering. Robust positionering och navigering kommer istället att kräva stöttning från andra sensorer som fungerar i en telekrigsstörd miljö. För att uppnå detta är det bra att använda en metod som ger god relativ noggrannhet som sedan kan låsas upp mot kända punkter i terrängen för att på så sätt åstadkomma global positionering med hög noggrannhet. Genom att hitta och följa stabila och väldefinierade punkter, s.k. *landmärken*, från de stödjande sensorerna kan ett förbättrat positions- och orienteringsestimat beräknas. Denna metod för att navigera utifrån tidigare okända landmärken kallas ofta *Simultaneous Localization And Mapping (SLAM)*.

Under de senaste två decennierna har forskning om SLAM varit ett mycket aktivt forskningsområde inom robotik. Med SLAM avses tekniker för att skapa en karta över ett område samtidigt som systemet lokaliserar sig själv relativt den skapade kartan. Med andra ord löser tekniken både positionerings- och karteringsproblemet parallellt. Eftersom kartan byggs upp efter hand är metoderna inte beroende av tidigare kunskap om omgivningen. Eftersom det alltid finns en viss måtosäkerhet hos sensorer är probabilistiska modeller det vanligaste tillvägagångssättet för SLAM.

En utförlig genomgång av olika metoder för SLAM finns i [9]. Nedan följer en kortfattad beskrivning av de tekniker som för närvarande studerats och använts inom projektet.

### 2.2.1 EKF-SLAM

Ett av de mer populära tillvägagångssätten för SLAM är att representera omgivningen i en tillståndsrymd med diskret tid med hjälp av en filterlösning kallat *Extended Kalman Filter (EKF)*. I EKF-SLAM-lösningar representeras fordonets position tillsammans med landmärken, som är distinkta punkter i en omgivning, i en tillståndsvektor som uppdateras i varje tidssteg med sensordata från t.ex. kameror, lidar eller radar. En *rörelsemodell* används för att bestämma hur den nuvarande tillståndsvektorn förhåller sig till tillståndsvektorn i nästföljande tidssteg. I varje tidssteg beräknas en optimal lösning för fordonets position och alla landmärkens position, endast baserat på det föregående tidssteget. Därmed kan beräkningarna i varje tidssteg kraftigt reduceras, till skillnad mot om alla tidssteg skulle ha tagits hänsyn till. Osäkerheten representeras av ett flertal täthetsfunktioner kopplade till rörelsemodellen, sensorobservationerna samt tillståndsvektorn. I modellen antas att genom att rekursivt uppdatera medelvärdet och kovariansmatriserna i dessa fördelningsfunktioner erhålls en optimal lösning för hur fordonet och dess omgivning förhåller sig till varandra.

Traditionella EKF-SLAM-lösningar har dock ett antal inbyggda begränsningar:

- Mängden beräkningar ökar kvadratisk med antalet landmärken, vilket gör det till en beräkningstung algoritm om många landmärken används.
- Kartan och identifierade landmärken använder en extern referens, oftast fordonets startpunkt. I och med att osäkerheten kring fordonets startposition ökar allteftersom

fordonet rör sig kommer nya landmärken till slut uppfattas som väldigt osäkra och tillföra fel vid rörelseuppdateringar.

- När fordonet rör sig uppdateras positionsestimatet enligt en rörelsemodell som matas med data från odometrisensorer. Odometrar har oftast låg noggrannhet, vilket gör att positionsfelet snabbt kan öka.
- EKF-SLAM skapar en rigid lösning för varje tidssteg som inte kan korrigeras, om ett fel uppstår i ett tidssteg kan det inte korrigeras i ett senare.

### 2.2.1.1 Robocentrisk EKF-SLAM

Många ansatser har studerats för att finna en lösning på begränsningarna i traditionell EKF-SLAM. En av dessa ansatser är att utnyttja en *robocentrisk* EKF-SLAM-lösning. I den robocentriska lösningen representeras fortfarande omgivningen i en tillståndsrymd med diskret tid, men skiljer sig från den traditionella lösningen på tre punkter:

- Genom att utnyttja lokala subkartor begränsas mängden använda landmärken, vilket minskar mängden beräkningar.
- Alla positioner och landmärken utgår från fordonets nuvarande position och inte en extern referens, vilket gör att den totala osäkerheten i varje subkarta minskar.
- Integrationen av odometridata görs först efter att fordonets position och landmärken har uppdaterats med nya data från övriga sensorer, vilket minskar risken för fel.

Med dessa förbättringar av den robocentriska EKF-SLAM-lösningen är det möjligt att minska det totala positioneringsfelet avsevärt.

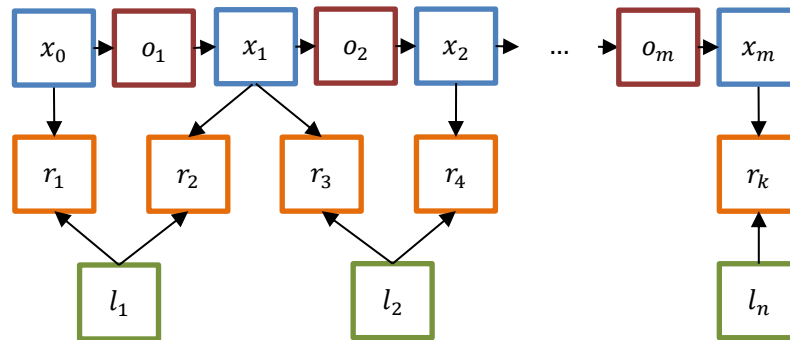
### 2.2.2 Smoothing and Mapping - SAM

Trots flera vidareutvecklingar för att komma tillrätta med begränsningarna av filterlösningar som EKF-SLAM-lösningen kvarstår dock flera problem. Ett av de största problemen är att fel som införts i ett tidssteg aldrig kan åtgärdas i ett senare tidssteg. För att komma tillrätta med denna problematik har ett helt annat tillvägagångssätt baserat på faktorgrafer, kallat *Smoothing And Mapping* (SAM), tagits fram.

Istället för att endast bygga vidare på det föregående tidsstegets lösning räknar SAM-algoritmerna ut en optimal lösning för nuvarande och alla föregående tidssteg i varje uppdatering. Nackdelen med detta tillvägagångssätt är att det leder till att fler beräkningar måste göras per tidssteg. Dock har moderna optimeringsmetoder och urvalsstrategier gjort att SAM-algoritmer idag är mycket beräkningseffektiva och kan vara ett alternativ till filterbaserade lösningar [10].

I SAM-algoritmer konstrueras det ursprungliga SLAM problemet i form av en faktorgraf som exemplifieras i figur 6. I figuren betecknas fordonspositionen i det  $i$ :e tidssteget  $x_i$ , rörelsen mellan tidssteg  $i$  och  $i + 1$  betecknas som  $o_{i+1}$  medan den  $k$ :e inmätningen av det  $j$ :e landmärket  $l_j$  betecknas  $r_k$ .

För att genomföra effektiva beräkningar beskrivs faktorgrafen i form av en *informationsmatrix*, se figur 7, där fordonspositionerna och landmärkesfaktorerna placeras längs diagonalen. Faktorer som beskriver relationerna mellan dessa, d.v.s. rörelsen mellan två positioner och inmätningen av landmärken från en position, placeras utanför diagonalen på fordonspositionernas och landmärkenas respektive rad och kolumn. Effektiviteten i SAM-lösningar erhålls genom att utnyttja den glesa matrisstruktur som uppstår i en faktorgraf allteftersom ett fordon rör sig genom ett område. För att bibehålla denna struktur över tid måste dock matrisen omorganiseras med jämna mellanrum. Denna process är i allmänhet mycket tidskrävande och beräkningstung, men den senaste tidens prestandaförbättringar för SAM-lösningar medför att det nu kan ske mycket effektivt.



Figur 6. Faktorgraf representation av SLAM problemet. I figuren betecknar  $x_i$ , fordonets position i det  $i$ :e tidssteget,  $o_{i+1}$  rörelsen mellan tidssteg  $i$  och  $i + 1$ ,  $l_j$  det  $j$ :e landmärket och  $r_k$  är den  $k$ :e inmätningen av ett landmärke.

	$x_0$	$l_1$	$x_1$	$l_2$	$x_2$
$x_0$		$r_1$	$o_1$		
$l_1$	$r_1$		$r_2$		
$x_1$	$o_1$	$r_2$		$r_3$	$o_2$
$l_2$			$r_3$		$r_4$
$x_2$			$o_2$	$r_4$	

Figur 7. Schematisk bild över hur en faktorgraf kan se ut i matrisform. I figuren betecknar  $x_i$ , fordonets position i det  $i$ :e tidssteget,  $o_{i+1}$  rörelsen mellan tidssteg  $i$  och  $i+1$ ,  $l_j$  det  $j$ :e landmärket och  $r_k$  är den  $k$ :e inmätningen av ett landmärke.

### 2.2.2.1 GT-SAM

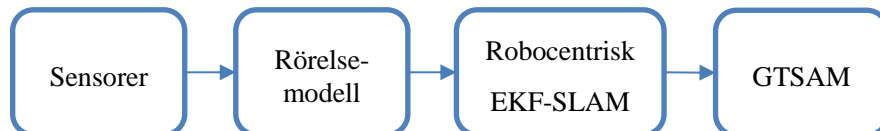
En metod för att göra denna omorganisering kallas *Incremental SAM* (ISAM) som inkrementellt strukturerar om informationsmatrisen vartefter nya fordonspositioner och landmärken läggs till.

Figur 7 visar att samtliga faktorer utanför diagonalen återkommer två gånger, en gång ovanför diagonalen och en gång nedanför. Något förenklat kan grundidén i ISAM beskrivas genom att endast ett av dessa element behålls för att minska mängden beräkningar. Detta görs genom att applicera en rotation, s.k. *Givens-rotation*, på de berörda raderna och kolumnerna. ISAM-algoritmen utnyttjar också det faktum att nya observationer oftast bara är beroende av de senaste positionerna och landmärkena. I matrisen i figur 7 finns till exempel inga beroenden (element utanför diagonalen) mellan  $x_0$  eller  $l_1$  och  $x_2$ . Detta medför att endast en begränsad mängd element i den nedre högra delen av matrisen behöver uppdateras och betydande effektiviseringar och beräkningsvinster erhålls för algoritmen. En utförlig beskrivning av ISAM-algoritmen finns i [11].

ISAM är implementerad i mjukvaruramverket *Georgia Tech Smoothing And Mapping* (GT-SAM). Utöver effektivare beräkningar av positioneringslösningar för ett fordon möjliggör ramverket även återigenkänning av tidigare besökta platser och positionering av flera samverkande fordon.

### 2.2.3 Beskrivning av LiDAR-baserat positioneringssystem

Det LiDAR-baserade positioneringssystemet som studerats i projektet kombinerar en robocentrisk filter-lösning med GT-SAM för att utnyttja fördelarna med de olika algoritmerna. I systemet sköter det robocentriska filtret den lokala positioneringen samt registrerar och återkopplar landmärken till tidigare tidssteg medan GT-SAM används för att förfina positioneringen och hitta en global positioneringslösning. Detta tillvägagångssätt medför att omorganisering av informationsmatrisen i GT-SAM och beräkningen av en ny lösning inte behöver ske i realtid. Därmed kan de största bristerna med SAM-algoritmen undvikas. En schematisk skiss över informationsflödet i positioneringssystemet visas i figur 8.



Figur 8. Illustration av informationsflödet från sensor till färdig positioneringslösning.

I positioneringssystemet används en LiDAR för detektion av landmärken. Fördelen med denna sensortyp är att den, till skillnad från kameror, har en mycket hög avståndsupplösning, vilket medför att många olika landmärken kan användas. Exempel på sådana landmärken är träd, stolpar och plana ytor och hörn på byggnader. Ytterligare en fördel med LiDAR är att den ger en detaljrik bild av omgivningen, vilket i ett senare skede kan användas för visualisering, ordergivning och beslutsstöd.

En IMU-sensor utgör grunden för positioneringssystemets rörelsemodell. I rörelsemodellen integreras accelerometer- och gyrodata för att skapa en skattning av fordonets hastighet och positionsförändring mellan två efterföljande tidssteg. Systemet har även möjlighet att utnyttja satellitnavigeringssystem (GNSS) för att ytterligare förfina hastighetsskattningen.

## 2.3 Spaning och sensorplanering

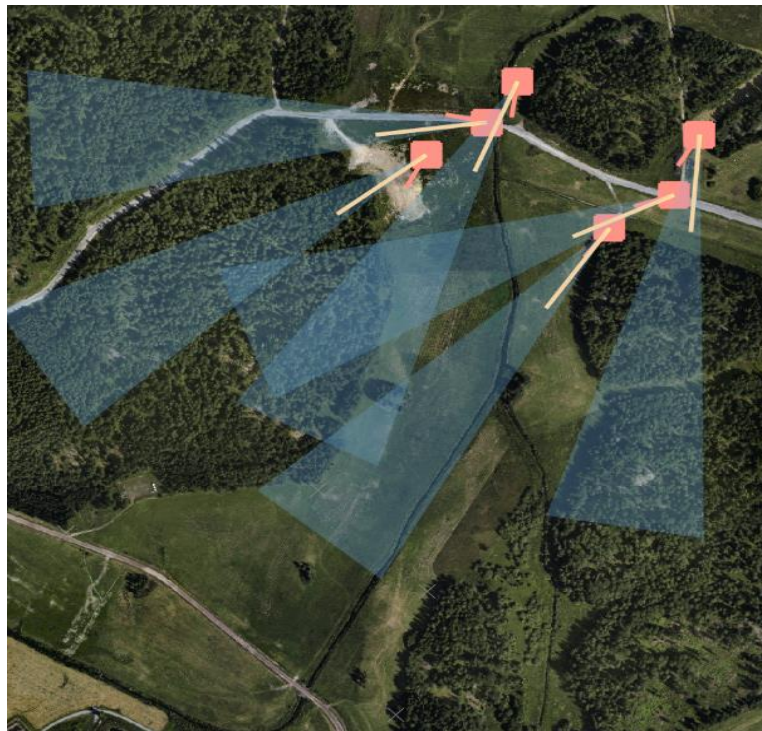
I projektet har ett scenario studerats där två stridsfordonsplutoner utrustade med spanings-sensorer framrycker i en småbruten terräng, se figur 9. I detta scenario är det av stor vikt att potentiella hot upptäcks så snabbt som möjligt för att snabbt kunna göra inmätning och klassificering av dessa. I en komplex situation med flera hot och ett begränsat antal sensorer krävs det att resurserna används på bästa möjliga sätt för att spaningsuppgiften ska lösas så bra som möjligt.

Komplexiteten är stor då den innehåller flera olika målvariabler och bivillkor:

- Terrängvariation, som påverkar sikten.
- Viktning mellan att följa och upptäcka nya hot.
- Inmätningprecision, t.ex. för indirekt bekämpning.
- Följa spaningsteknik enligt reglemente.
- Sensorprestanda.

För enskilda sensoroperatörer kan detta vara en svår uppgift att lösa och speciellt i en strids-situation där tidsaspekten från upptäckt till möjlig bekämpning är kritisk. Därför har det inom projektet studerats hur automatiska sensorplaneringsfunktioner och sensorsamverkan kan stödja sensoroperatören och höja den totala lägesuppfattningen. Med sensorsamverkan menas att information såsom position, hastighet, måldetektioner och målspar kan utbytas mellan plattformarna.



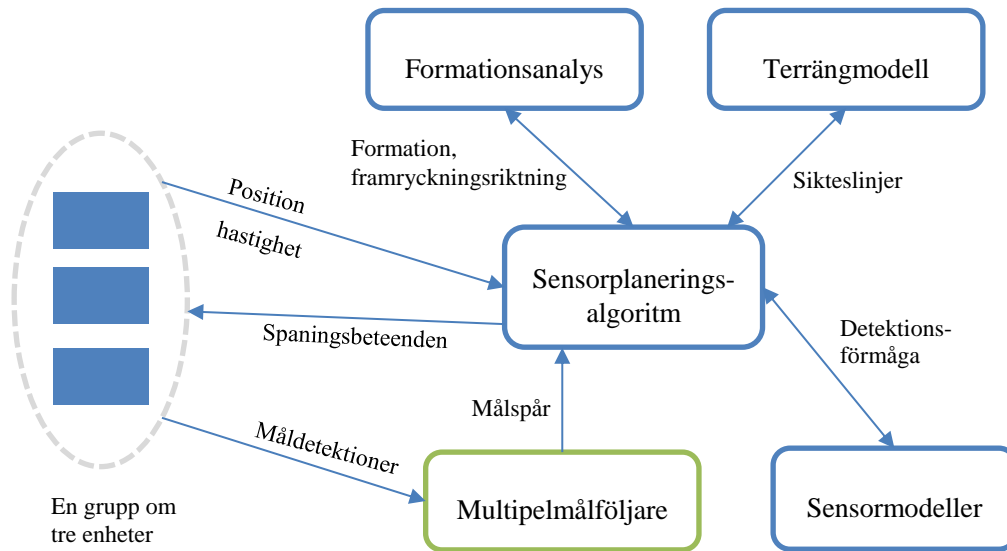


Figur 9. Exempelscenario för att illustrera sensorplanering. Två stridsfordonsplutoner framrycker söderut i stridskolonnformation.

Arbetet inom spaning och sensorplanering påbörjades i föregående projekt, *Intelligent spaning* [12]. Föregående arbete var i huvudsak inriktat mot stillastående sensorplattformar där grundfunktioner implementerades i ett sensorplaneringsramverk. I ramverket hanterades sensorer och plattformar, där plattformarna kan vara olika typer av sensorbärare: t.ex. stridsfordon, master och RPA:er. Ramverket har nu byggts ut för att hantera grupperingar av plattformar, rörliga plattformar, plattformar med flera sensorer, spaningsreglemente för stridsfordonsplutoner och integration med andra simuleringsmjukvaror tillgängliga vid FOI.

Sensorplaneringen är utvecklad utifrån två grundbeteenden: söka efter mål och följa upptäckta mål. I varje planeringssteg beräknas den bästa kombinationen av sensorer som skall utföra respektive beteende för att maximera spaningsnyttan. Beräkningen av kombinationer nyttjar olika mått som beskriver hur mycket ny information (positioneringsprecision) som tillförs och till vilken kostnad [13]. I litteraturen är detta ett vanligt angreppssätt på sensorplaneringsproblem och detta baseras på informationsteori [14]. Ett exempel kan vara att ett mål har upptäckts av en spaningssensor, som ger god bäringsinformation, men med sämre avståndsinformation. I detta fall skulle krysspejling av en annan sensor tillföra mer information då osäkerheten i avståndsled för den ena sensorn kompenseras av vinkelupplösningen hos den andra. Genom att utnyttja detta kan en god precision i målposition snabbt erhållas och sedan kontinuerligt upprätthållas. Problemet är uppställt som ett optimeringsproblem som löses genom att använda en auktionsalgoritm [15]. Auktionsalgoritmen beräknar den optimala kombinationen av sensorer till mål med avseende på möjlig ny information och till vilken kostnad.

I det studerade scenariot framrycker de två plutonerna i olika formationer: stridskolonn, stridstriangel och på stridslinje. Plutonens formation och tänkta framryckningsriktning beräknas kontinuerligt. Position och hastighet för varje enhet i plutonen används för att göra beräkningarna. För respektive formation riktas spaningssensorerna i olika riktningar för att maximera observationsområdet. I ramverket har funktionalitet för grupper av plattformar och formationer implementerats, vilket nu möjliggör analys av spaningsprestanda för större scenarier och där gällande spaningsreglemente följs. I figur 10 illustreras schematiskt en översikt av sensorplaneringsramverket och dess ingående komponenter.



Figur 10. En schematisk översikt av sensorplaneringsramverket och dess ingående delkomponenter. Information som går från gruppen är position och hastighet för varje enhet och måldetektioner i gruppens samtliga sensorsystem. Multipelmålföljaren är en extern delkomponent som skickar beräknade målspår till sensorplaneringsalgoritmen. Utifrån den information som skickas beräknas de spaningsbeteenden som maximalt förbättrar lägesbilden.

### 3 Sensorsystemet MSP

Projektet har vidareutvecklat experimentsystemet Multisensor plattform (MSP) genom att integrera nya sensorer och algoritmer. Sedan tidigare bestod MSP-systemet av en visuell kamera och en IR-sensor monterade på ett vridbord (eng. Pan-Tilt-Unit, PTU). Mjukvara för detekktion, styrning och målföljning har funnits i experimentsystemet. Projektet har kompletterat MSP-systemet med det LiDAR-baserade positioneringssystemet, som beskrivs i avsnitt 2.2.3. Detta möjliggör att detekktion, positionering och målinmätning nu kan ske helt automatiskt.

För närvarande består MSP-systemet av följande delar:

- En långvågig IR-sensor (LWIR), Flir SC655
- En visuell kamera, Pointgrey ZBR2
- En LiDAR, Velodyne VLP-16
- En IMU, Xsens Mti-710
- Ett vridbord, FLIR PTU-D300E ISM
- En dator för bearbetning av LiDAR- och IMU-data, positionering samt kommunikation med PTU (PC-1)
- En dator för inhämtning och bearbetning av visuella och IR-data (PC-2)

LiDAR-sensorn består av 16 stycken roterande laseravståndsmätare med en våglängd på 905 nm och en uteffekt på 2 mW per laser, vilket gör lidarn ögonsäker. Räckvidden hos lidarn är 100 meter och rotationshastigheten kan ändras mellan 5-20 Hz. Utöver ovan nämnda komponenter ingår även diverse mindre delsystem för bland annat synkronisering. Figur 11 visar det vidareutvecklade MSP-systemet monterat på ett fordon.

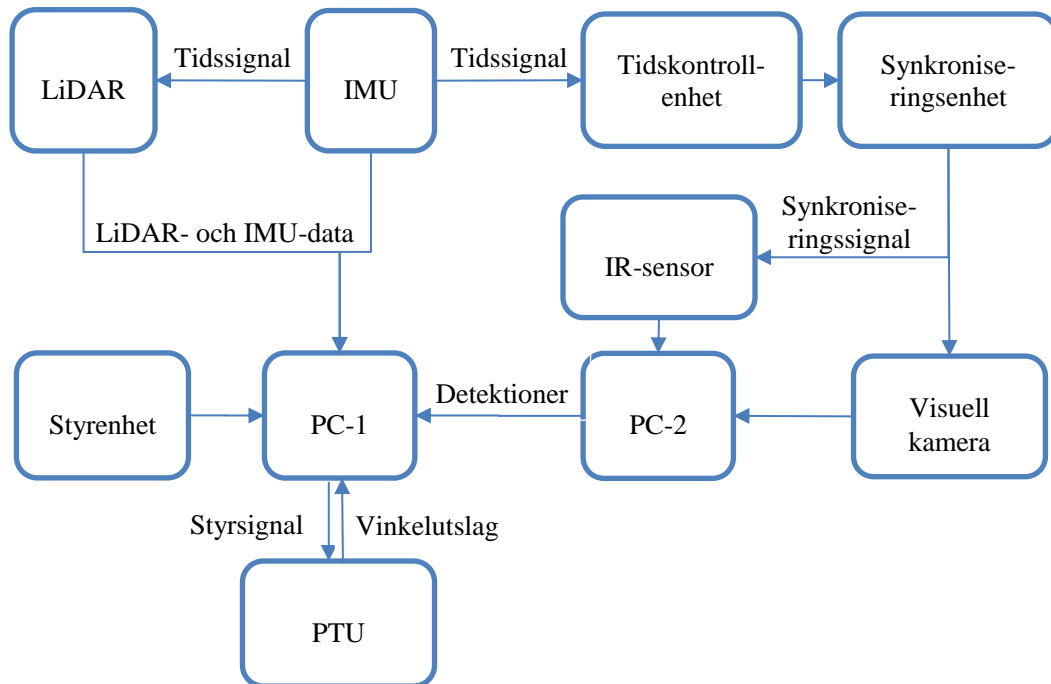


Figur 11. MSP-systemet tillsammans med det LiDAR-baserade positioneringssystemet monterat på ett mätfordon.

Det LiDAR-baserade positioneringssystemet ger en positionsangivelse för fordonet, medan vinkelutslag från vridbordet skapar en länk mellan fordonet och de olika sensorerna. Detta medför att MSP-systemet i dagsläget automatiskt kan positionsbestämma ett mål under färd. En förutsättning för att lyckas med detta är att tidssynkroniseringen mellan sensorerna är tillförlitlig, d.v.s. det måste gå att avgöra vilken position, tittriktning samt detekktion som

kommer från samma tidssteg. Att skapa en gemensam tidsbas mellan samtliga sensorer och att få den att vara korrekt har varit ett mycket viktigt arbete för att kunna analysera målinmättningsförmågan med systemet.

I MSP-systemet görs detta genom att IMU-enhetens interna klocka används som tidsbas i hela systemet. IMU-enheten skickar en tidsangivelse i form av en fyrkantsvåg, samt ett UDP-meddelande, till både LiDAR-sensorn och tidskontrollenheten. Tidskontrollenheten och synkroniseringsenheten ansvarar för att IR-sensorn och den visuella kameran har samma tidsreferens som övriga systemet. En principskiss över hur de olika komponenterna kommunicerar med varandra visas i figur 12.



Figur 12. Principskiss för tidsynkronisering av experimentsystemet MSP. IMU-enheten skickar en tidsangivelse till både LiDAR-sensorn och tidskontrollenheten. Tidskontrollenheten och synkroniseringsenheten ansvarar för att IR-sensorn och den visuella kameran har samma tidsreferens som övriga systemet.

## 4 Resultat för signalbehandlingsalgoritmerna

I projektet har olika signalbehandlingsalgoritmer utvecklats för att förbättra spaningsförmågan hos sensorplattformar med styrbara EO/IR-sensorer. I detta avsnitt redovisas hur dessa signalbehandlingsalgoritmer kan användas som operatörsstöd.

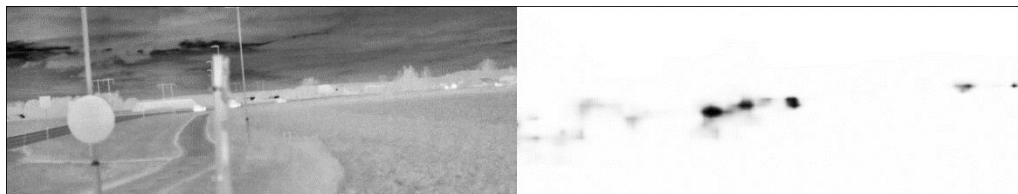
### 4.1 Operatörsstöd med måligenkänning

Neuronnäten som beskrivits i avsnitt 2.1 har implementerats i ramverket TensorFlow<sup>3</sup> [16] och tränats på en kraftfull arbetsstation med fyra grafikkort av modell NVIDIA Geforce GTX 1080 Ti. Till viss del kunde befintlig kod från Google (TensorFlow Object Detection API<sup>4</sup>) utnyttjas, men betydande modifieringar har gjorts, bl.a. för att hantera klassen ”Ignorera”, dvs. regioner av bilderna som inte ska klassificeras under träningen.

IR-bilderna har ursprungligen 14 bitars dynamik, men denna reducerades till 8 bitar inför träningen. Vid dynamikkompressionen användes en egenutvecklad algoritm som kombinerar lokal Laplacefiltrering [17] med kontrastbegränsad lokal histogramutjämning (CLAHE) [18]. Denna metod bibehåller lokal kontrast i bilden och kan hantera stora temperaturskillnader (t.ex. mellan kall himmel och soluppvärmd vägbanan).

För båda algoritmerna användes nätverksarkitekturen ResNet-50 [19] som särdragsextraktor. Detta nät med 50 lager har 25 miljoner parametrar och är därmed betydligt mindre än de nät som leder tävlingen COCO Detection<sup>5</sup>, men har ändå goda prestanda. Den begränsade mängden träningsdata var avgörande för val av nätverk.

Särdragsextraktorn *förtränades* som del av ett nätverk för semantisk segmentering vars arkitektur liknar RetinaNet, med skillnaden att nätet för varje bildpunkt skattar sannolikheten att punkten tillhör ett objekt, se figur 13. Annoterade träningsdata för detta problem (ca 100 000 bilder) genererades automatiskt genom förgrundsextraktion, figur 14.



Figur 13. Semantisk segmentering. Till höger visas för varje bildpunkt den skattade sannolikheten att punkten tillhör ett objekt från de aktuella klasserna.



Figur 14. Träningsexempel för semantisk segmentering. Annoteringen till höger är automatiskt genererad genom förgrundsextraktion av objekt i rörelse.

Förträningen innebär att när träningen av objekt-detektorn påbörjas har särdragsextraktorn redan meningsfulla parametervärden. Det har nämligen visat sig att de representationer som

<sup>3</sup> [www.tensorflow.org](http://www.tensorflow.org)

<sup>4</sup> [github.com/tensorflow/models/tree/master/research/object\\_detection](https://github.com/tensorflow/models/tree/master/research/object_detection)

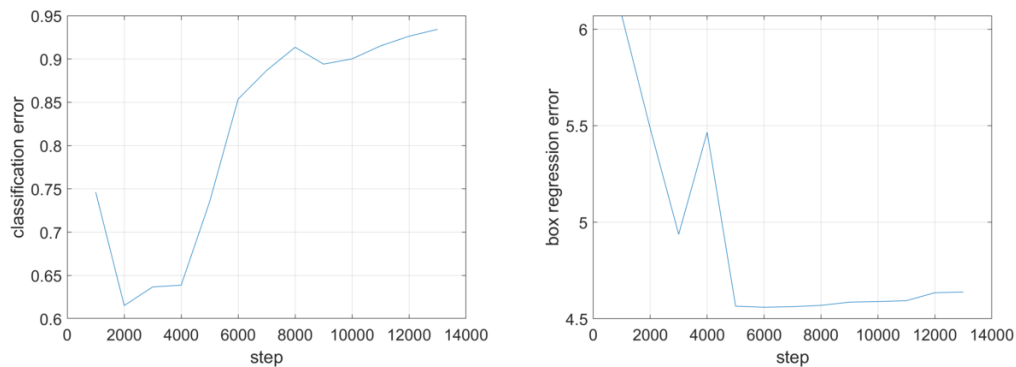
<sup>5</sup> [cocodataset.org/#detection-leaderboard](https://cocodataset.org/#detection-leaderboard)

skapas i de första lagren i ett djupt neuronät som tränats att klassificera stora mängder data är generellt användbara för många olika uppgifter [20,21].

Efter förträningen av särdragsextraktorn tränades hela nätverken med 2686 manuellt annoterade bilder från 70 sekvenser. För validering användes återstående 1194 bilder från 23 sekvenser. Uppdelningen mellan tränings- och valideringsdata baserades på GPS-positioner för kameran vid inspelningen. Kriteriet här var att sekvenser från de båda kategorierna måste vara separerade med minst 500 meter. Valideringssekvenser valdes så att deras närmaste granne måste vara en träningssekvens. Detta för att tillse att tränings- och valideringsdata fick liknande fördelning av miljöer (så att valideringsdata exempelvis inte bara kom från innerstaden i Linköping).

Några typiska resultat visas i figur 16. Här markeras alla objekthypoteser med skattad sannolikhet på minst 20 %. RetinaNet har genomgående större förmåga att detektera små objekt, i synnerhet i bilder med låg kontrast, men har också något högre falsklarmsintensitet än Mask R-CNN.

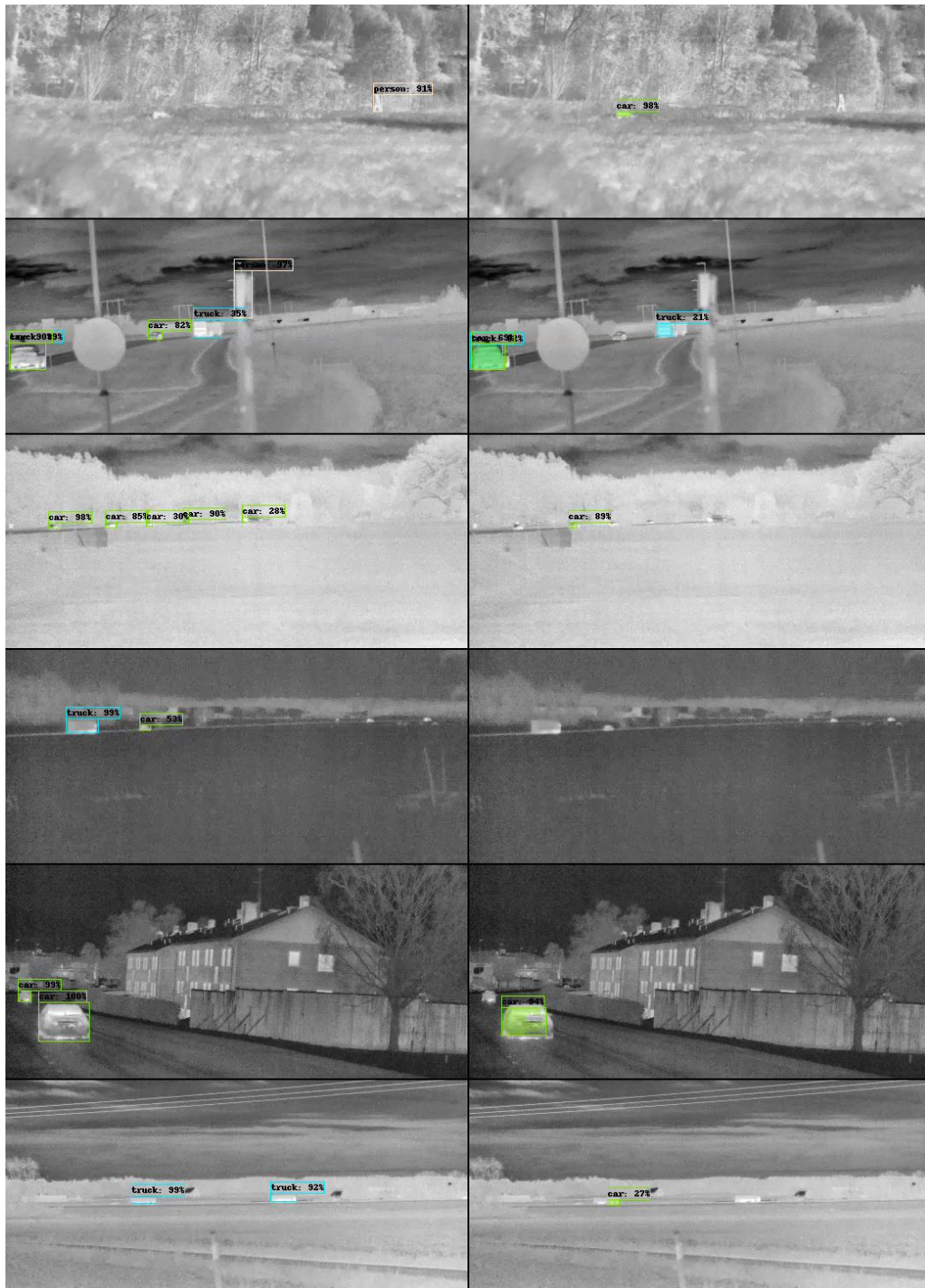
Valideringsresultaten visar tydligt att mängden träningsdata är för litet. Efter ett relativt litet antal iterationer börjar valideringsfelet växa, se Figur 15, vilket är ett säkert tecken på överträning.



Figur 15. RetinaNet. Klassificerings- och regressionsfel för valideringsdata som funktion av antalet träningssteg.

Mer systematiska valideringar av algoritmerna har inte gjorts då mängden valideringsdata som har tagits fram är för begränsat. Vad som kan konstateras är dock att redan med en relativt liten uppsättning (några tusen) annoterade träningsexempel kan neuronätsbaserade algoritmer tränas att detektera och klassificera objekt i termisk IR-video. Objekt från klasserna med flest träningsexempel (bilar, lastbilar och personer) kan med rätt hög säkerhet korrekt klassificeras om storleken överstiger 20 pixlar i huvudriktningen. Detta gäller i synnerhet om klassificeringen kombineras med målföljning. De mera sällsynta klasserna är svårare, så där behövs flera träningsexempel. Det förekommer en del falsklarm, i synnerhet från objekt som framställts av människor såsom byggnader, stolpar och skyltar; även här krävs mer data.

RetinaNet har fördelen att vara betydligt snabbare än Mask R-CNN och klarar lätt bearbetning av video i realtid. Resultaten visar också att bättre klassificeringsprestanda uppnås med RetinaNet, i synnerhet för små objekt. Det är dock möjligt att förbättra Mask R-CNN genom att använda top-downbearbetning och multipla skalor på samma sätt som i RetinaNet för såväl detektions- som klassificeringssteget.



Figur 16. Exempel på bearbetningsresultat för valideringsdata. Vänster: RetinaNet. Höger: Mask R-CNN. Bilderna har beskurits något för tydligare visualisering.

## 4.2 Analys av positioneringsprestanda

För att analysera positioneringsprestanda har två försök genomförts. I det första försöket har ett enskilt fordon färdats efter en bana varefter det totala positioneringsfelet mäts upp. I det andra försöket har ett mål, som i detta fall har varit en person som detekterats i IR-sensorn, mätts in från två självpositionerande fordon varefter målets position har bestämts i ett gemensamt koordinatsystem.

Mätningar för utvärdering av positioneringssystemet har utförts i tre olika terrängtyper: bebyggelse, betäckt terräng och öppen terräng. Vid mätningarna har det LiDAR-baserade-positioneringssystemet monterats på taket på ett fordon. För försöken med målinmätning av en person har MSP-systemet, som har en IR-sensor, använts.

I försöken stötts skattningen av systemets hastighet för närvarande med GPS-mätningar. Arbete för att skatta hastigheten utan GPS har skett parallellt med framtagning av resultaten. I avsnitt 4.2.1 presenteras kvalitativa resultat för egenpositioneringen. Kvantitativa resultat för positioneringen med GPS-stöd samt GPS-fri hastighetsskattning presenteras i avsnitt 4.2.2. Resultaten för positionering av målet som detekterats med IR-sensorn presenteras i avsnitt 4.2.3.

#### 4.2.1 Kvalitativa resultat av egenpositionering

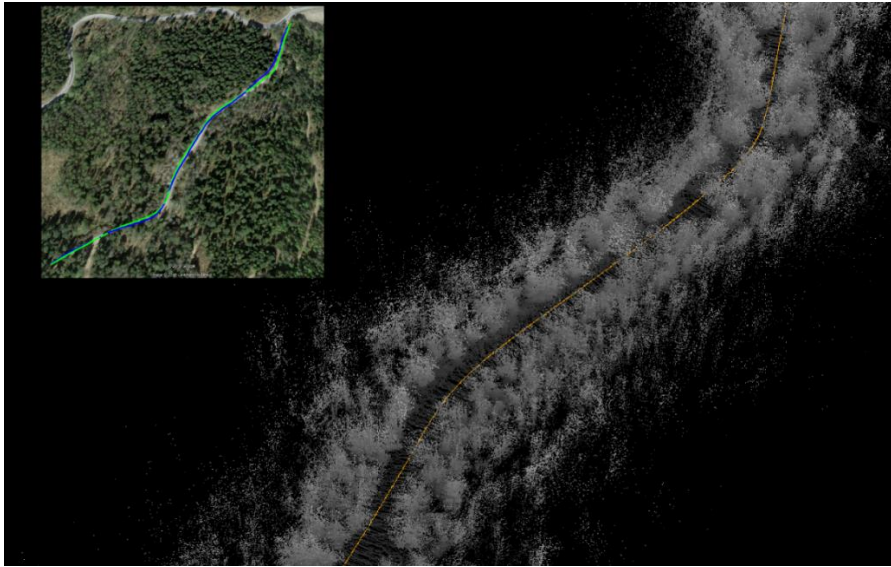
Resultatet för ett enskilt fordon som färdats längs en bana har analyserats. Egenpositionen har skattats samt kartan som samtidigt skapas av algoritmen (se kap 2.2). Detta resultat kan utvärderas genom att jämföra den uppmätta banan med GPS-data, med skattningar från andra SLAM-algoritmer, genom att använda positioneringssystemet vid flera repeterade körningar på samma sträcka, eller med ett ortofoto (t.ex. Google Earth som nedan).

Exempel på egenpositionering för bebyggelse, betäckt terräng och öppen terräng visas i figur 17, figur 18 samt figur 19. De uppmätta banorna i GPS (blå linjer) har en osäkerhet på flera meter. Figuren visar att den skattade banan ibland ligger utanför den väg som finns i ortofotot. Även ortofotot kan ha ett noggrannhetsfel. Figuren illustrerar ett kvalitativt resultat: systemet fungerar i olika terräng med en rimlig osäkerhet.



Figur 17. Erhållen 3D-karta från egenpositioneringen i bebyggelse. De grå punkterna visar LiDAR-sensorns uppmätta punkter i 3D, inlagd i det globala koordinatsystemet. Den skattade banan är utritad i orange. Inflikat i det övre högra hörnet är samma georefererade bana (grön linje, motsvarar orange linje i 3D-kartan) samt uppmätta GPS-data (blå linje) som är överlagrad ett ortofoto från Google Earth (observera att ortofotot är något inaktuellt).





Figur 18. Erhållen 3D-karta från egenpositioneringen i skog. De gråa punkterna visar LiDAR-sensors uppmätta punkter i 3D, inlagd i det globala koordinatsystemet. Den skattade banan är utritad i orange. Bilden som är inflikad i övre vänstra hörnet är samma georefererade bana utritad (grön linje) samt GPS-data som uppmätts (blå linje), vilket överlagrats på ett ortofoto från Google Earth.

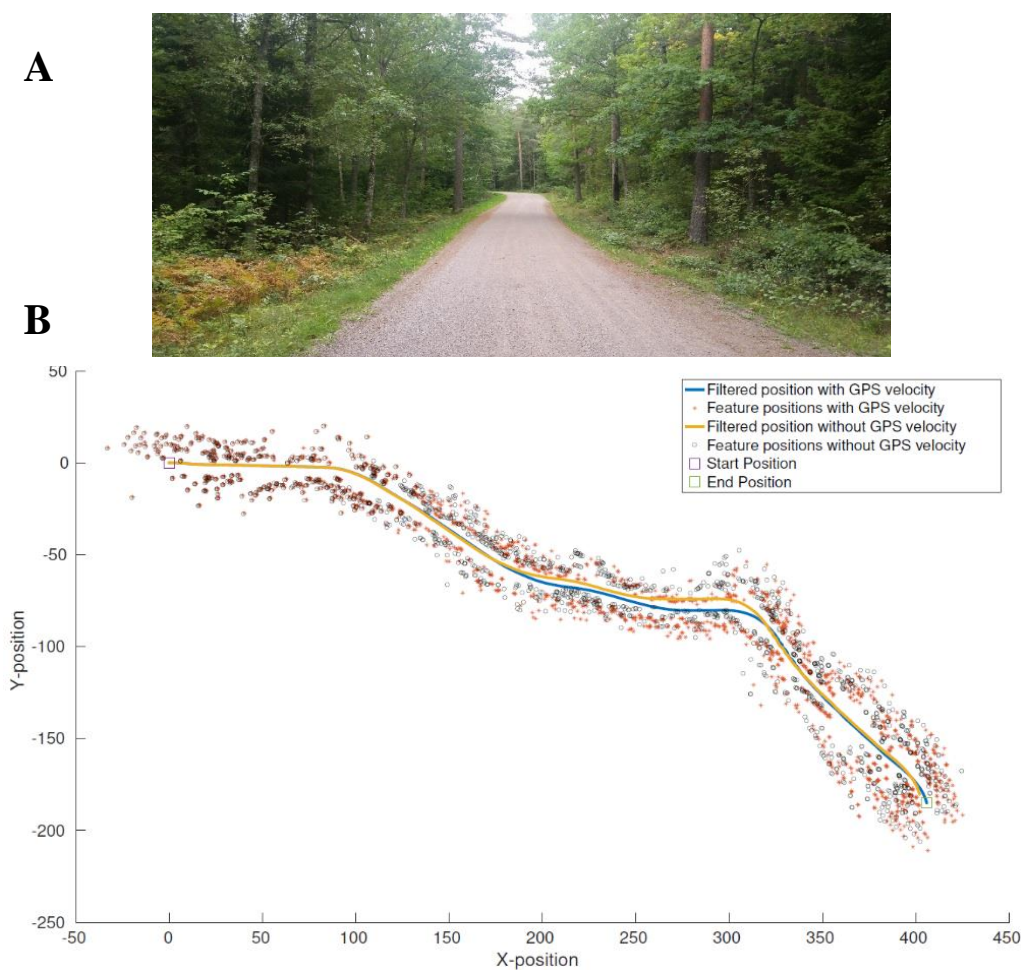


Figur 19. Erhållen 3D-karta från egenpositioneringen i öppen terräng. De gråa punkterna visar LiDAR-sensors uppmätta punkter i 3D, inlagd i det globala koordinatsystemet. Den skattade banan är utritad i orange. Inflikat i det nedre vänstra hörnet ses samma georefererade bana (grön linje) samt uppmätt GPS-data (blå linje) som är överlagrad på ett ortofoto från Google Earth. Längst uppe till höger visas en in-zoomning i 3D-kartan på den övre delen av banan.

#### 4.2.2 Jämförelse mellan GPS-stödd och GPS-fri positionering

I detta avsnitt presenteras en jämförelse mellan GPS-stödd positionering och positionering helt utan hjälp av GPS. Jämförelsen görs i två olika terrängtyper, längs skogsväg (figur 20) samt runt ett huskvarter (figur 21). Längs med skogsvägen finns gott om trädstammar som har använts som landmärken, vilket ger en god skattning även utan GPS-stödd hastighet. I scenariot med huskvarteret går det sämre, då en stor överskattning av hastigheten görs i den senare delen av banan med det GPS-fria systemet. Detta beror på att i denna del (övre delen av banan, se också figur 17) finns det inga byggnader eller träd och därför har algoritmen svårt att hitta tydliga landmärken.

Kvantitativa resultat kan ses i tabell 1, där värdena visar jämförelser mellan GPS-stödd och GPS-fri positionering. De kvantitativa resultaten visar att i miljöer med många landmärken kan skillnaden mellan GPS-stödd och GPS-fri positionering vara mindre än 1 % i medelvärde. Den stora utmaningen är naturligtvis att bli oberoende av GPS även i det svårare fallet med öppen terräng utan tydliga landmärken.

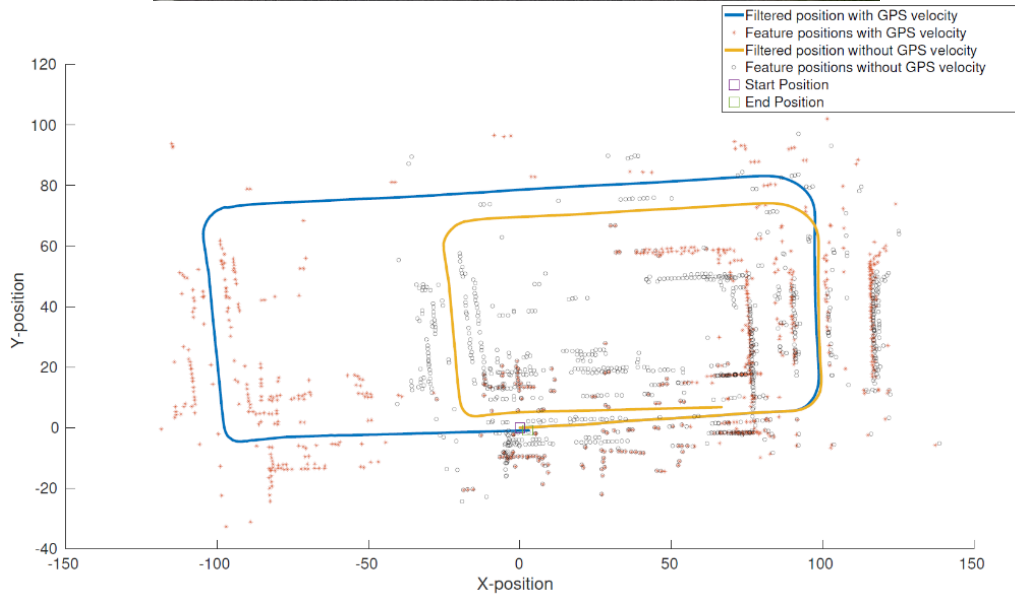


Figur 20. Mätning på skogsväg. A: representativt foto från miljön. B: Den uppmätta banan och landmärken med och utan GPS-stödd hastighetsskattning.

A



B



Figur 21. Mätning på huskvarter. A: foto från banan, på "utsidan" av kvarteret (motsvarar övre långsidan i nedersta bilden (B)). B: Den uppmätta banan och landmärken med och utan GPS-stödd hastighetsskattning. Notera att på "utsidan" av banan får positioneringssystemet svårt att hitta tydliga landmärken, vilket gör att den GPS-fria mätningen måste använda tröghetsmätningar som orsakar drift och gör att hastigheten överskattas.

Tabell 1. Kvantitativ skillnad mellan GPS-stödd och GPS-fri positionering. Riktningen på koordinatsystemet (x, y) är godtyckliga, men felmåtten ger en fingervisning på det absoluta felets storlek.

	Skogsterräng	Öppen terräng
Medelfel per meter i x-led	0.0087	0.0603
Medelfel per meter i y-led	0.0072	0.0086
Drift per meter i x-led	0.0081	0.1176
Drift per meter i y-led	0.0098	0.0142

#### 4.2.3 Positionering av utpekat mål

I det andra försöket har ett mål, i detta fall en person som detekteras med IR-sensorn, mätts in från två självpositionerande fordon. Därefter har målets position skattats i ett gemensamt koordinatsystem. Endast ett positioneringssystem har utvecklats i projektet så positioneringssystemet har därför mätt in det stillastående målet vid två separata mätningar.

Inmätning av ett mål ställer mycket högre krav på det egna systemets positioneringsnoggrannhet jämfört med positionsbestämning av endast det egna systemet. Detta beror på att

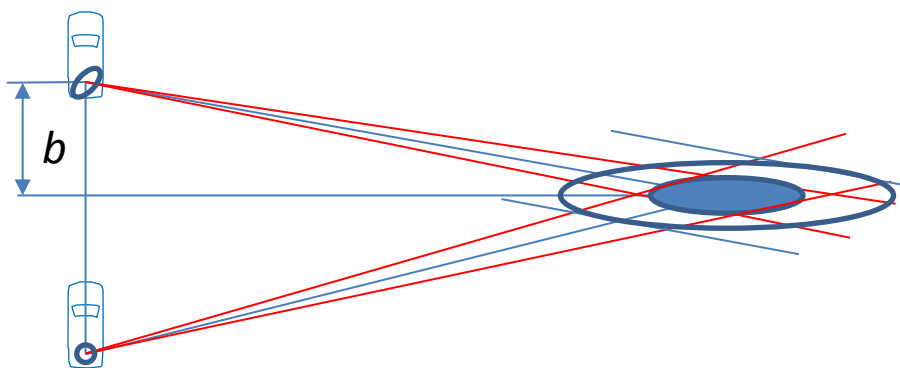
inmätningen av ett mål skapar en hävarm till målet som förstärker den vinkelosäkerhet som finns hos positioneringssystemet.

Vid utpekning av mål används begreppet målutpekningfel (eng. Target location error, TLE) som är skillnaden mellan den faktiska och uppmätta positionen för ett mål. Target location error (TLE) delas ibland in i sex klasser (CATS) som sträcker sig från bästa (CAT I) till sämst (CAT VI), vilket visas i tabell 2. Precisionsstyrda vapen och ammunition (som är beroende av GNSS) är bara så noggrann som den inriktning koordinater ges till dem. Utmaningen är därför att mäta in målet med låg TLE.

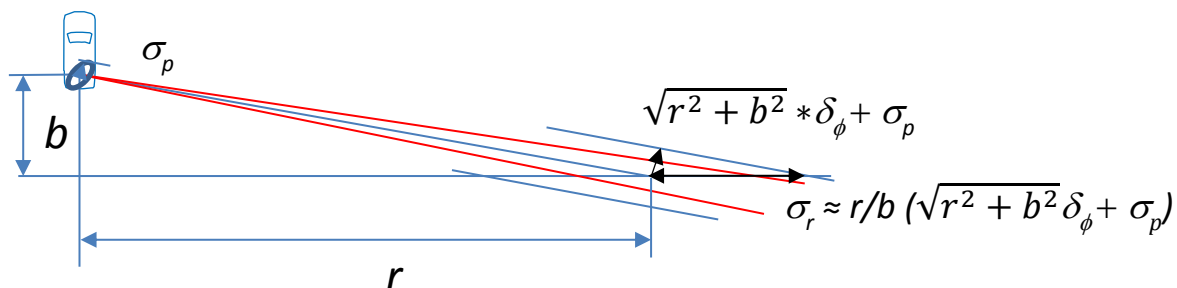
Tabell 2. Målutpekningfel indelat i sex klasser (CATS).

CAT I	CAT II	CAT III	CAT IV	CAT V	CAT VI
0-6 m	7-15 m	16-30 m	31-91 m	92-305 m	>305 m

Det finns ett antal tekniker för att öka noggrannheten för målinmätning, se [22], och en av dessa tekniker är triangulering. I dessa försök har målinmätningen förbättrats genom triangulering av mål från två olika system med ett gemensamt koordinatsystem. Då en liten vinkelosäkerhet är avgörande för målinmätningen kommer särskild hänsyn tas till hur denna osäkerhet kan minskas genom samverkande positioneringssystem. I figur 22 illustreras hur en gemensam inmätning sker från två fordon.



Figur 22. Principskiss för gemensam inmätning med vinkelmätande sensor. Den lilla ellipsen representerar approximativ osäkerhet för enbart vinkelosäkerhet och den större inkluderar även effekter av positionsosäkerhet. För att lättare illustrera problemet har vinkelosäkerheten överdrivits i figuren.

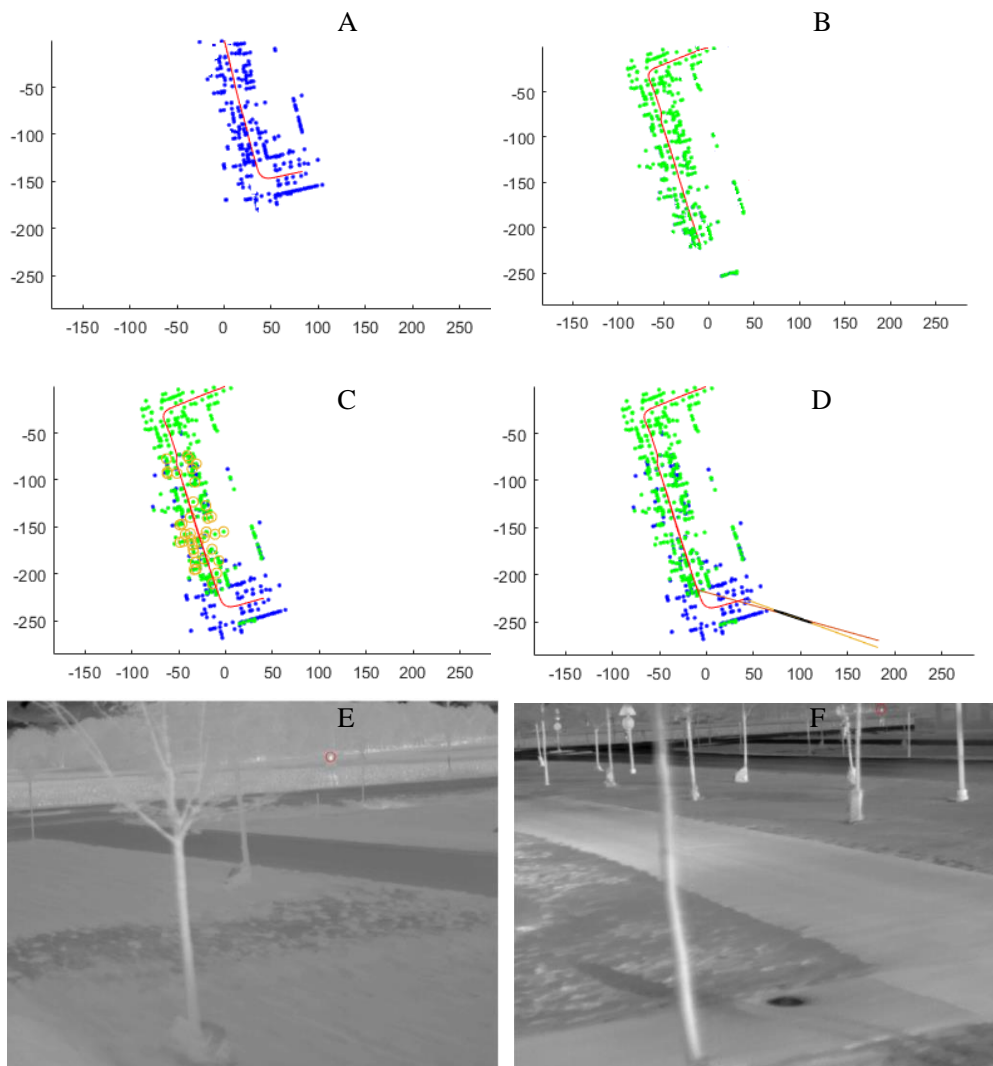


Figur 23. Skiss för överslagsberäkning av avståndosäkerhet givet en baslinje  $b$  som är mindre än avståndet  $r$ .

En tumregel för hur osäkerheten beror av baslinjen mellan fordonen och vinkelosäkerheten kan härledas ur geometrin i figur 23 där baslinjen  $b$  och avståndet till målet  $r$  bildar en likformig triangel med motsvarande triangel för mätosäkerheten och den sökta avståndosäkerheten. Om inget systematiskt fel (bias) antas och en målposition med 90 % säkerhet (Circular error, CE90) skall uppnås krävs att vinkelfelet i sigma ( $\sigma$ ) är en faktor 2.146 mindre.

Resultat av tumregeln för en baslinje om 100 m illustreras i tabell 3. Som referens kan nämnas att för en IR-sensor med 25 graders öppningsvinkel och  $640 \times 480$  pixlar i upplösning ger en 0.5 pixels detektion (1-sigma) ett vinkelfel på 0.345 milliradianer. För att uppnå CE90 krävs därmed att vinkelfelet understiger 0.74 milliradianer ( $2.146 \times 0.345$ ).

Syftet med försöket är att se med vilken noggrannhet ett mål kan inmätas givet vinkelmätning mot målet från två fordon, som är självpositionerande (figur 24C). De två fordonen positionerar sig relativt kartan med hjälp av uppmätta landmärken (figur 24A), och relateras till varandra genom gemensamma landmärken som associeras (Figur 24B). Från var sitt läge upptäcks samma mål med IR-sensorn (figur 24D). Vinklarnas skärningspunkt pekar då ut målet.

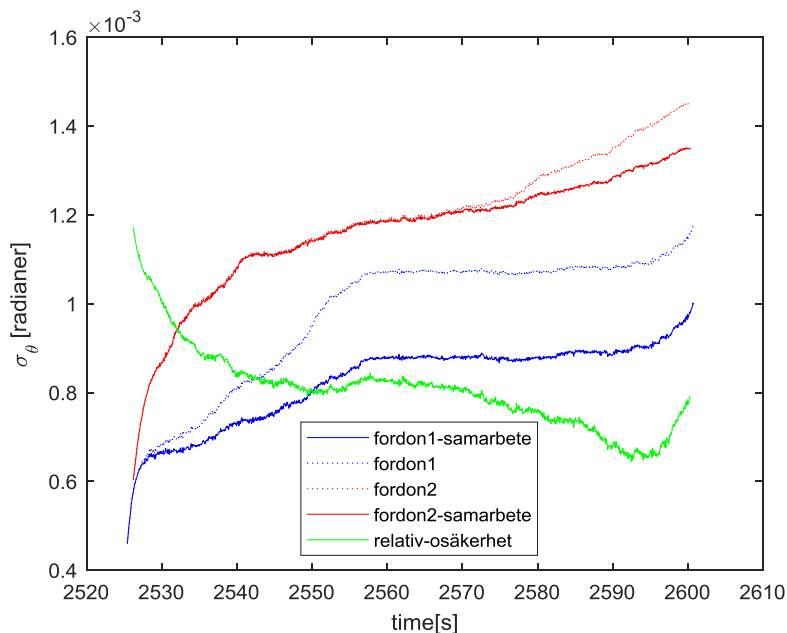


Figur 24. Två fordon med positioneringssystemet, vars landmärken är markerade med grönt (system 1) och blått (system 2). En person detekteras från två positioner, vilka är markerade i figur E och F. Figur A och B: de båda systemens orelaterade koordinatsystem. Figur C: Koordinatsystemet för system 2 relateras till system 1 genom ett antal gemensamma och automatiskt detekterade landmärken markerade med gula ringar. Figur D: De båda systemen detekterar och mäter in ett mål gemensamt, osäkerheten för inmätningen markeras med en svart färgad elips. Skattningen av målets position är skärningen av vinklarna från respektive sensor. Noggrannheten (svart ellips) beror på systemens egenpositioneringar, associationen mellan dem, osäkerheten i sensors orientering, samt osäkerheten i detektionerna.

Tabell 3. Nödvändig vinkelnoggrannhet för baslinje  $b = 100$  m (baserat på enkel tumregel och ingen statistisk bias).

	Avstånd till objekt		Avstånd till objekt		Avstånd till objekt		Övre gräns acceptabelt fel (m)
	500 m		1000 m		1500 m		
	Milliradianer	Streck	Milliradianer	Streck	Milliradianer	Streck	
CAT I	1.10	1.10	0.28	0.28	0.12	0.12	6
CAT II	2.74	2.75	0.70	0.70	0.31	0.31	15
CAT III	5.48	5.50	1.39	1.39	0.62	0.62	30
CAT IV	16.63	16.68	4.22	4.23	1.88	1.89	91

I figur 25 illustreras GT-SAM algoritmens skattning av vinkelosäkerheten för de två fordonens position samt den relativa vinkelosäkerheten.



Figur 25. Vinkelosäkerhet för de två fordonen före och efter samarbete via delade landmärken. Den relativa osäkerheten är mindre än de individuella osäkerheterna utom i början tack vare de delade landmärkena.

Genom att jämföra de heldragna linjerna med de streckade röda respektive blå linjerna i figuren ser man att vinkelosäkerheten minskar när två positioneringssystem samverkar. Då målet trianguleras utifrån de båda positioneringssystemen är det den relativa vinkelosäkerheten som kommer att avgöra den slutgiltiga inmätningosäkerheten för målet. Då denna osäkerhet understiger 1.1 milliradianer uppfylls, tillsammans med två oberoende detektioner med en baslinje på 100 m och 0.25 pixels noggrannhet, kraven för inmätning av CAT I i tabell 3. Förbättring i noggrannhet blir inte samma för båda fordonen, då skattningen beror på olika geometrier i färdväg samt fordonens olika läge.

#### 4.2.4 Diskussion

Resultat har presenterats från det LiDAR-baserade positioneringssystemet och dessa visar att systemet kvalitativt fungerar för att skatta den egna positionen, för att förbättra positionering genom att dela mätningar mellan plattformar, samt för att peka ut upptäckta mål på en

karta. Skattningen av noggrannheten i det nuvarande systemet är sannolikt något optimistisk, vilket i huvudsak beror på att LiDAR-sensorn har mycket god mätnoggrannhet. En felkälla som inte tagits hänsyn till i modellen är felassociationer av landmärken, vilket lokalt kan ge upphov till betydligt större fel.

Jämförelsen mellan ett helt GPS-fritt positioneringssystem och det nuvarande systemet med GPS-stödd hastighetsskattning visar också på landmärkenas betydelse. Avsaknad av landmärken gör att tillståndsskattningen driver och kan leda till stora fel om information endast kommer från tröghetssensorn.

Mer avancerade sätt att associera kända landmärken med nya mätningar, samt att kunna hitta landmärken i alla miljöer, även vid öppna fält, är den främsta utmaningen för systemet. Om informationen från nuvarande sensorer, främst LiDAR-sensorn, är otillräcklig finns möjligheten att stödja systemet med ytterligare sensorer, t.ex. en visuell kamera eller en IR-sensor. Befintliga visuella kameror eller IR-sensorer på markfordon kan då vara bra att använda. Det är också tänkbart att mer avancerade LiDAR-sensorer som har flera lasersensorer (horisontella linjer), t.ex. 32- eller 64-linjer i stället för de nuvarande 16-linjerna.

En styrka med positioneringssystemet är att det kommer vara oberoende av GPS. Men detta betyder inte att GPS inte kan utnyttjas när det finns tillgängligt. Modellen tillåter att GPS-information kan komplettera övrig information, vilket medför ytterligare förbättringar avseende positioneringsnoggrannhet. Därmed är det också möjligt att i framtiden utnyttja systemet för att upptäcka GPS-störningar, d.v.s. när det är stor diskrepans mellan GPS-information och övriga sensorsignaler (IMU) och geometriinformation (uppmätta landmärken).

### 4.3 Förbättrad spaningsförmåga med automatisk sensorplanering

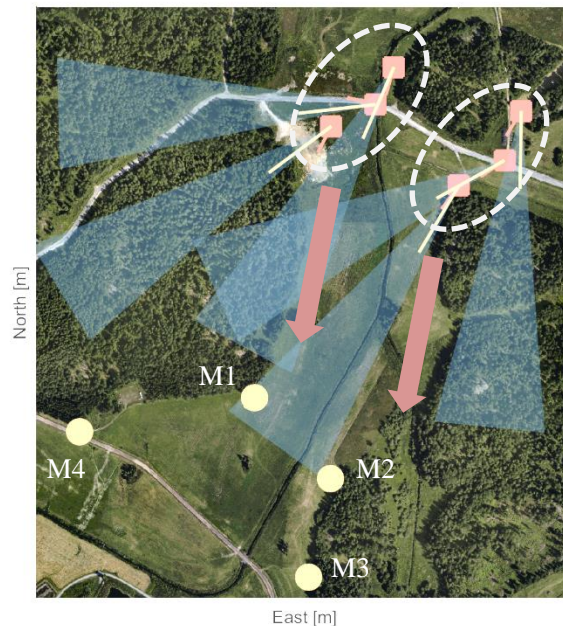
För att utvärdera hur automatisk sensorplanering och sensorsamverkan kan ge en ökad spaningsförmåga har ett militärt scenario använts vid utvecklingen och utvärderingen av sensorplaneringsramverket. Eftersom hela utvärderingen kräver återkoppling från sensorer har utvärderingen gjorts i simuleringsmiljön MSSLab (Multisensorsimuleringslab) [23]. I simuleringsmiljön är det givna scenariot implementerat där två plutoner, om tre stridsfordon, framrycker i terrängen, se figur 26. Varje stridsfordon är utrustad med en styrbar IR-sensor, vilken används för att spana i området omkring plutonerna och i framryckningsriktningen. Totalt finns därmed sex sensorer tillgängliga. Sensorerna kan samverka genom att kommunicera måldetektioner, egenposition och hastighet. En central målföljare används för att beräkna målspar (position och hastighet) utifrån måldetektionerna från IR-sensorn. Dessa målspar kan sedan användas för att skapa en gemensam lägesbild för samtliga stridsfordon.

I utvärderingen har vi behandlat olika aspekter av höjd spaningsförmåga. De aspekter vi har valt att studera mera är hur olika spaningsstrategier påverkar följande spaningsfaktorer:

- målsparprecision över tid
- tid till upptäckt av mål
- målsparens livslängd

En spaningsstrategi definieras av olika parametrar där vi har valt att utvärdera:

- antalet sensorer som tillåts att låsa på mål
- sensorernas svephastighet
- planeringshorisont



Figur 26. Framtaget scenario med två plutoner i ett område om ca  $1000 \times 1000 \text{ m}^2$  i Kvarn. Plutonerna framrycker söderut och de röda pilarna markerar fordonets orientering. Varje stridsfordon är utrustad med varsin styrbar spanings sensor vars momentana synfält illustreras med en blå triangel. Längs med skogsridåerna är fyra mål utplacerade: M1, M2, M3 och M4.

Den första strategiparametern beskriver det antalet sensorer, per pluton, som sensorplaneringsalgoritmen har tillgång till för att automatiskt rikta mot mål för att öka målsårsprecisionen. Desto fler sensorer som finns tillgängliga desto fler mål kan hållas uppdaterade med tillräcklig precision. En sensors svephastighet beskriver hur snabbt en sensor tillåts att svepa per tidsenhet, grader/s. En hög svephastighet möjliggör att snabbt rikta sensorn i en given riktning. Planeringsintervallet beskriver hur ofta en beräkning av den bästa fördelningen av spaningsbeteende till sensorer görs, se kapitel 2.3. Under ett planeringsintervall ändras inte en sensors spaningsbeteende, utan det är då konstant.

Då vi från simuleringsmiljön vet var målen är placerade i terrängen kan de estimerade målsårspåren jämföras med den sanna positionen. Målsårspåren byggs upp av tillståndsvektorer, som beskriver målets position och hastighet över tid. Utifrån dessa kan det beräknade felet från sann position användas för beräkning av målsårsprecision. Målsårsprecision beräknas som roten av det kvadratiske medelvärdesfelet, *Root Mean Square Error* (RMSE). RMSE beskriver spridningen i avstånd, mätt i meter, från målets sanna position. De två övriga tidsfaktorerna extraheras från målföljaren. Genom att utnyttja Monte-Carlo-simulering fås bättre förståelse för hur de beräknade faktorerna är fördelade med medelvärde och standardavvikelse.

I tabell 4 är de olika spaningsstrategierna definierade. Utifrån dessa strategier utvärderas sedan de beräknade spaningsfaktorerna. En grundstrategi, G, är framtagen med standardparametrar för att underlätta vid utvärdering. Strategierna A, B, C är sedan varianter av G där olika värden på respektive parameter varieras.

Genom att låta fler sensorer följa specifika mål kan sensorplaneringsalgoritmen behålla en hög målsårsprecision speciellt när fler mål har upptäckts. I tabell 5 ser vi att redan när en sensor tillåts att låsa mål utanför sitt ordinarie spaningsområde, A2, så ökar målsårsprecisionen med en faktor 10. När inga sensorer finns tillgängliga att låsa på mål, A1, så spanar sensorn endast i den riktning som bestäms av aktuell formation, dvs. rakt fram, åt vänster eller åt höger. Utifrån detta resultat kan vi dra slutsatsen att spaningsfaktorerna förbättras



desto fler sensorer som tillåts låsa på mål, men ur resultatet erhålls redan en god förbättring med en till två tillgängliga sensorer.

Svephastigheten borde påverka samtliga spaningsfaktorer eftersom en hög hastighet möjliggör att snabbt rikta sensorn i en given riktning. Med hög svephastighet kan en sektor snabbt sökas av och tid mellan växling av mål, att följa, kan minimeras. I tabell 6 kan vi dra slutsatsen att en hög svephastighet främst påverkar målsparprecisionen där både medelvärdet och standardavvikelsen minskar. För en mänsklig sensoroperatör är det svårt att tolka IR-bilder när svephastigheten är för hög, men för en algoritm gör det ingen skillnad.

*Tabell 4. Definition av de spaningsstrategier som används under utvärderingen av spaningsförmåga. Strategi G är grundstrategin som är satt med framtagna standardparametrar. Strategierna A, B, C är varianter av grundstrategin G. Parametrar som varierar är markerade med kursiv stil.*

<i>Strategi</i>	<i>Sensorer att låsa på mål per pluton</i>	<i>Svephastighet</i>	<i>Planeringsintervall</i>
<i>G</i>	2	80 grader/s	3 s
<i>A</i>	<i>A1: 0</i> <i>A2: 1</i> <i>A3: 2</i> <i>A4: 3</i>	80 grader/s	3 s
<i>B</i>	2	<i>B1: 10 grader/s</i> <i>B2: 40 grader/s</i> <i>B3: 80 grader/s</i> <i>B4: 180 grader/s</i>	3 s
<i>C</i>	2	80 grader/s	<i>C1: 0.10 s</i> <i>C2: 1 s</i> <i>C3: 3 s</i> <i>C4: 5 s</i>

Ett kort planeringsintervall torde ge bästa resultat för att så ofta som möjligt beräkna bästa spaningsbeteende för samtliga sensorer. Utifrån tabell 7 kan vi dra slutsatsen att så inte är fallet. Istället är det bättre att beräkna spaningsbeteenden som gäller ett kort tidsintervall framåt i tiden. Därmed kan predikterad framtida information tas med i beräkningen, t.ex. positionsförändringar och terrängförhållanden. Målsparprecisionen minskar inte i absoluta tal, men spridning mellan iterationerna blir mindre vilket gäller även för tidsfaktorerna.

*Tabell 5. Resultat för antalet sensorer som automatiskt tillåts att låsa på mål över 100 Monte-Carlo iterationer. Talen är medeltalet och standardavvikelsen över alla iterationer och sedan över alla mål. Vid jämförelse erhålls bästa resultat då fler sensorer tillåts att låsa på mål, som är utanför deras ordinarie spaningssektor.*

	<i>Målsparprecision [m]</i>	<i>Tid till upptäckt [s]</i>	<i>Livslängd [s]</i>
<i>A1</i>	42.4 ± 33.2 m	200 ± 56 s	70 ± 40 s
<i>A2</i>	4.16 ± 6.11 m	164 ± 41 s	124 ± 40 s
<i>A3</i>	2.62 ± 2.25 m	156 ± 39 s	137 ± 39 s
<i>A4</i>	1.69 ± 1.16 m	146 ± 27 s	150 ± 28 s

Tabell 6. Resultat för flera svephastigheter över 100 Monte-Carlo iterationer. Talen är medeltalet och standardavvikelsen över alla iterationer och sedan över alla mål. Vid jämförelse erhålls bästa förändring från B2 till B3 där B4 verkar vara det bästa alternativet, dvs. 180 grader/s.

	Målsårsprecision [m]	Tid till upptäckt [s]	Livslängd [s]
B1	2.68 ± 3.17 m	155 ± 42 s	135 ± 42 s
B2	2.45 ± 2.95 m	142 ± 20 s	154 ± 20 s
B3	1.97 ± 1.51 m	146 ± 28 s	150 ± 28 s
B4	1.90 ± 0.86 m	144 ± 26 s	152 ± 26 s

Tabell 7. Resultat för flera planeringsintervall över 100 Monte-Carlo iterationer. Talen är medeltalet och standardavvikelsen över alla iterationer och sedan över alla mål. Vid jämförelse erhålls bästa förändring från C1 till C2 där C3 verkar vara det bästa alternativet i detta fall, dvs. tre sekunders planeringsintervall.

	Målsårsprecision [m]	Tid till upptäckt [s]	Livslängd [s]
C1	2.48 ± 4.00 m,	161 ± 25 s	133 ± 24 s
C2	1.90 ± 2.14 m	151 ± 17 s	146 ± 18 s
C3	1.97 ± 1.51 m	146 ± 13 s	150 ± 13 s
C4	2.10 ± 2.25 m	148 ± 14 s	147 ± 14 s

Resultaten visar att sensorer som tillåts att styras automatiskt kan höja spaningsförmågan. Genom att de styrs automatiskt och data behandlas av algoritmer fås en högre målsårsprecision, eftersom högre svephastighet är möjligt. Med automatisk sensorplanering kan vi även visa att det ger en högre spaningsförmåga eftersom när ett hot har detekterats av någon sensor väljs den sensor som ger mest ökning av målprecisionen och övriga sensorer kan fortsätta spana. Detta bidrar till ett mer resurseffektivt användande av tillgängliga sensorer. Resultaten visar även att planering för ett tidsintervall framåt i tiden kan ge bättre spaningsförmåga eftersom terränginformation och positionsprediktion av plattformar och mål tillsammans kan utnyttjas för att ta hänsyn till sensorers upptäcksförmåga och skydd sikt. Längden av tidsintervallet beror på komplexiteten i ett visst scenario och i detta givna simuleringsfall gav tre sekunder det bästa resultatet.

## 5 Fortsatt arbete

De framtagna signalbehandlingsalgoritmerna har redan idag bra funktionalitet, men utvecklingen inom området är mycket snabb. Fokus på arbetet har varit signalbehandling för stridsfordon, men algoritmerna som beskrivs i rapporten är generella och kan användas även användas av övervakningssystem, fartyg, helikoptrar, RPAS (Remotely Piloted Aircraft System, fjärrstyrt flygsystem), flygplan etc. Vissa av algoritmerna behöver dock tränas om eller modifieras för att kunna användas i andra scenarier.

Arbete med automatisk måligenkänning börjar ge bra resultat, men mer träningsdata behövs föra att förbättra resultaten. I detta arbete har vi fokuserat på civila objekt för att resultat ska kunna spridas, men arbete bör fortsätta med militära objekt för att se hur bra algoritmerna kan känna igen sådana objekt. Forskningen inom objektigenkänning är mycket intensiv för tillfället och utvecklingen inom området är mycket snabb.

Arbetet med positioneringssystemet börjar ge bra resultat. I fortsatt arbete bör vi studera hur mycket bättre positioneringssystemet blir om man kombinerar information från en LiDAR med geografisk terränginformation som t.ex. Lantmäteriets Nationella höjdmmodell. I öppen terräng får positioneringssystemet svårt att hitta tydliga landmärken med de nuvarande sensorerna. Därför bör fortsatt arbete studera om positioneringen kan stöttas med ytterligare sensorer, t.ex. en visuell kamera eller en IR-sensor. Befintliga visuella kameror eller IR-sensorer på markfordon kan då vara bra att använda. Positioneringen kan också förbättras genom terrängmatchning i LiDAR-mätningarna. Det är också tänkbart att mer avancerade LiDAR-sensorer som har flera lasersensorer (horisontella linjer), t.ex. 32- eller 64-linjer i stället för de nuvarande 16-linjerna kan användas.

Inom sensorplanering har lovande resultat framkommit under projektets gång. På konceptuell nivå krävs mer arbete för att få olika plattformar att effektivt samarbeta mot ett gemensamt mål. Nya maskininlärningsmetoder som t.ex. reinforcement learning bör studeras för att undersöka hur styrning och planering av rörliga sensorsystem kan förbättras.

## 6 Referenser

- [1] Degani, A. *Taming HAL: designing interfaces beyond 2001*, New York: Palgrave Macmillan, 2004.
- [2] Sarter, N. B., Woods, D. D. och Billings C. E. Automation surprises, G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics*, New York: Wiley, 1997.
- [3] Billings C. E., *Aviation Automation: The Search for a Human-Centered Approach*, Mahwah, NJ: Erlbaum, 1997.
- [4] Svenmarck, P. "Principles of human-robot coordination for improved manned-unmanned teaming", FOI Memo 1507, Linköping, 2005.
- [5] Krizhevsky, A., Sutskever, I., Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks", NIPS, 2012.
- [6] Ren, S., He, K., Girshick, R., Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks", NIPS, 2015.
- [7] He, K., Gkioxari, G., Dollár, P., Girshick, R., "Mask R-CNN", ICCV, 2017.
- [8] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., "Focal Loss for Dense Object Detection", ICCV, 2017.
- [9] Näsström, F., Bissmarck, F., Deleskog, V., Gustafsson, D., Hendeby, G., Karlholm, J., Nordlöf, J., Nygård, J., "Omvärldsanalys inom signalbehandlingsalgoritmer för EO/IR-sensorer", FOI-R--4287--SE, juni 2016.
- [10] Dellaert, F., Kaess, M., "Square Root SAM: Simultaneous localization and mapping via square root information smoothing", *The International Journal of Robotics Research* 25.12 (2006), 1181-1203.
- [11] Kaess, M., Ranganathan, A., Dellaert, F., "iSAM: Fast incremental smoothing and mapping with efficient data association." *Robotics and Automation*, 2007 IEEE International Conference on. IEEE, 2007.
- [12] Näsström, F., Bissmarck, F., Deleskog, V., Gustafsson, D., Hendeby, G., Karlholm, J., Nordlöf, J., Nygård, J., "Intelligent spaning – slutrapport", FOI-R--4148--SE, december 2015.
- [13] Hero, A.O., Castan, D., Cochran, D., Kastella, K., "Foundations and Applications of Sensor Management", Springer Publishing Company, Inc., 2010.
- [14] Shannon, C. E., "A Mathematical Theory of Communication", *The Bell System Technical Journal*, 1948.
- [15] Blackman S. S. och Popoli, R. "Design and Analysis of Modern Tracking Systems", Artech House, 1999.
- [16] Abadi, M., et al., "TensorFlow: A system for large-scale machine learning", OSDI, 2016.
- [17] Aubry, M., Paris, S., Hasinoff, S. W., Kautz, J., Durand, F., "Fast Local Laplacian Filters: Theory and Applications", *ACM Transactions on Graphics*, 33(5), art.no. 167, 2014.
- [18] Zuiderveld, K., "Contrast Limited Adaptive Histogram Equalization", *Graphics Gems IV*. Red. P. S. Heckbert. San Diego: Academic Press Professional, 1994.
- [19] He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition", CVPR, 2016.
- [20] Oquab, M., Bottou, L., Laptev, I., Sivic, J., "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", CVPR, 2014.
- [21] Razavian, A. S., Azizpour, H., Sullivan, J., Carlsson, S., "CNN Features off-the-shelf: an Astounding Baseline for Recognition", CVPR, 2014.
- [22] Internet: [https://armypubs.army.mil/ProductMaps/PubForm/Details.aspx?PUB\\_ID=1003069](https://armypubs.army.mil/ProductMaps/PubForm/Details.aspx?PUB_ID=1003069), besökt 2018-10-28.
- [23] Näsström, F., Allvar, J., Berggren, J., Bergström, D., Deleskog, V., Forsgren, R., Hedström, J., Lif, P., "Simulering av sensorsystem för sensorvärdering", FOI-R--4283--SE, juni 2016.