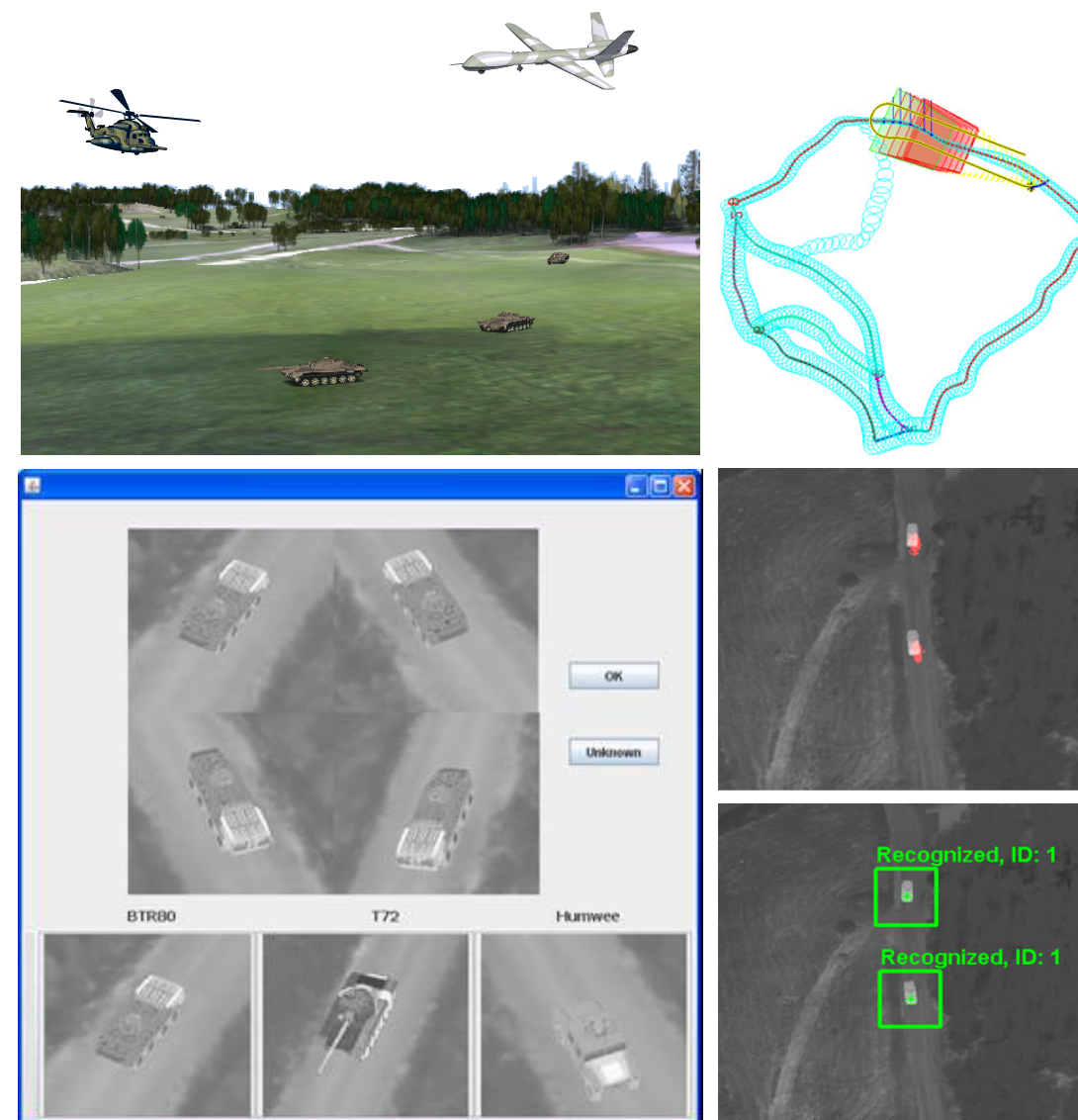


FREDRIK NÄSSTRÖM, MIRSAĐ CIRKIC, JOAKIM RYDELL,
PER SKOGLAR, KARL-GÖRAN STENBORG, MORGAN ULVKLO



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, Mirsad Cirkic, Joakim Rydell,
Per Skoglar, Karl-Göran Stenborg, Morgan Ulvklo

Autonoma funktioner för intelligenta operatörsstöd

Titel	Autonoma funktioner för intelligenta operatörsstöd
Title	Autonomous functions for operator supports
Rapportnr/Report no	FOI-R--2752--SE
Rapporttyp Report Type	Underlagsrapport Base data report
Sidor/Pages	31 p
Månad/Month	Mars
Utgivningsår/Year	2009
ISSN	ISSN 1650-1942
Kund/Customer	FMV
Forskningsområde Programme area	4. Sensorer och signaturanpassning
Delområde Subcategory	55 Flygteknik 55 Aeronautics
Projektnr/Project no	E3345
Godkänd av/Approved by	
FOI, Totalförsvarets Forskningsinstitut	FOI, Swedish Defence Research Agency
Avdelningen för Informationssystem	Information Systems
Box 1165	Box 1165
581 11 Linköping	SE-581 11 Linköping

Sammanfattning

Autonoma funktioner är ett projekt inom beställningen Flygteknik 06-08 med syfte att studera hur ett operatörsstöd kan förbättras genom att utnyttja sensordataanalys och autonoma funktioner. Det finns flera anledningar till varför man vill förbättra funktionaliteten hos ett operatörsstöd. Studier har visat att kapaciteten hos en sensoroperatör avtar markant till exempel då stora mängder sensordata ska behandlas eller då sensordata ska analyseras under en längre tid. För att stötta en operatör kan olika autonoma funktioner användas såsom autonom detektion, följning, återigenkänning, sensorstyrning och ruttplanering.

Projektet Autonoma funktioner har bidragit till att integrera dessa autonoma funktioner till MSSLab (MultiSensorSimuleringsLab). MSSLab är en simuleringsmiljö som är utvecklad på FOI för att möjliggöra värdering av sensorsystem i olika miljöer, väderförhållanden och tidpunkter.

I denna rapport ges en beskrivning av de autonoma funktionerna och hur dessa kan användas för att ge ett intelligent operatörsstöd. Scenariot som har studerats består av en UAV som flyger före en fordonskolonn för att spana efter hot och vid behov förse kolonnen med information så att den kan omplanera sin rutt. Fordonen i kolonnen utför en patrullering i en småbruten terräng med både civila och fientliga fordon.

Nyckelord: Autonoma funktioner, operatörsstöd, detektion, målföljning, återigenkänning, sensorstyrning

Summary

The aim of the project Autonomous functions is to study how operator supports can be improved by using sensor data analysis and autonomous functions. There are several reasons to improve the performance of operator support system. Studies have shown that the performance of sensor operators decreases considerably if for example a lot of sensor data are to be processed or if sensor data are analysed for a long time. To support operators' different autonomous functions can be applied as detection, tracking, reidentification, sensor control etc.

The project Autonomous functions have contributed to the integration of autonomous functions to MSSLab (Multi sensor simulation laboratory). MSSLab is a simulation framework developed at FOI to make it possible to evaluate sensor systems in various environments, weather conditions and time periods.

This report presents some autonomous functions and how these can be used to create intelligent support systems. The scenario used in the simulations contain a UAV flying in front of a convoy to search for threats and if so report that to the convoy so it can choose another route. The vehicles in the convoy are doing a patrol in a terrain with both hostile and neutral vehicles.

Keywords: Autonomous functions, operator supports, detections, tracking, reidentification, sensor control

Innehållsförteckning

1	Inledning	7
2	Autonoma funktioner	9
2.1	Detektion	9
2.2	Målföljning	12
2.3	Återigenkänning	15
2.4	Ruttplanering och sensorstyrning	16
3	Simuleringsmiljön MSSLab	19
3.1	Terrängmodellen	19
3.2	Objektmodeller	21
3.3	Scensimulering	21
4	Autonoma funktioner för intelligenta operatörsstöd	23
4.1	Scenario	23
4.2	Användning av autonoma funktioner för intelligenta operatörsstöd	24
4.3	Resultat	29
5	Slutsatser och fortsatt arbete	30
5.1	Fortsatt arbete	30
6	Referenser	31

1 Inledning

I denna rapport studeras hur ett operatörsstöd kan förbättras genom att utnyttja sensordataanalys och autonoma funktioner. Ett spaningsscenario har använts för att studera detta, men metoderna är tillämpbara även inom övervakning och till viss del även bekämpning.

I de flesta applikationerna är perceptionskapaciteten hos det mänskliga synsinnet än så länge överlägset dagens signalbehandlingsalgoritmer. Det finns trots det många situationer då en ökad nivå av signalbehandling ombord på farkosten, sensorplattformen, skulle öka den totala förmågan för såväl bemannade som obemannade system.

Följande argument motiverar ökad grad av intelligent operatörsstöd:

- Den uthålliga perceptionsförmågan hos en sensoroperatör eller fototolk är begränsad och studier visar att kapaciteten avtar påtagligt efter redan 30 minuters arbete. Då många större UAV-system är kapabla till spanings- och övervakningstider på flera dygn, så kan inte ens skiftstjänstgöring hos personalen garantera att sensoroperatören hela tiden är alert. Målutpekningssystem med automatiserade spaningsmoder kan här avlasta operatören markant.
- En mänsklig operatör har en begränsad förmåga att genomföra flera uppgifter parallellt då varje uppgift kan kräva full uppmärksamhet under en viss tidsrymd och skifte mellan uppgifterna kräver en vidare överblick för att ge rätt kontext. Ett typexempel är en operatör som detekterar ett antal markmål, zoomar in på ett potentiellt mål och samtidigt förlorar de övriga utanför synfältet. I ett sådant scenario skulle ett snabbt multimålföljningssystem med ett väl utformat gränssnitt kunna höja systemprestanda påtagligt om det kombinerades med ett system för autonom sensorstyrning.
- Olika metoder för förändringsanalys, som kan detektera onormala signalvariationer, kan öka prestanda i repetitiva övervakningssituationer. Ett stort problem är att kunna hantera den stora variabiliteten i naturliga scener via existerande fysiska och empiriska metoder då dessa hittills har begränsad modelleringsförmåga. System för visualisering av tidigare inhämtade sensordata, projicerade för att efterlikna den aktuella flygningen, kan för en operatör nyttjas för att detektera förändringar. Vissa skillnader kan lyftas fram via automatiserade metoder, t.ex. detektion av nya hjulspår, även i en klottrig scen. En sådan uppgift är mycket tidsödande för en mänsklig operatör.
- Autonoma inisierings- och utpekningssystem som är kapabla att detektera, följa och geolokalisera multipla markmål skulle påtagligt hjälpa en sensoroperatör att kunna fokusera på relevanta delmängder av sensordata under tidskritiska uppdrag.
- Data från multi- och hyperspektrala sensorsystem kräver avancerad förbehandling innan de kan presenteras för det mänskliga synsinnet.

- I vissa UAV-system kan förmågan till laserutpekning och sensorstyrning påtagligt förbättras via målföljning och sensorstyrning ombord på plattformen eftersom tidsfördröjningar i kommunikationssystemen, SATCOM, idag är alltför hög. Detta gäller framförallt vid interkontinental kommunikation och är därmed rimligen inte ett prioriterat problem för den svenska försvarsmakten.

2 Autonoma funktioner

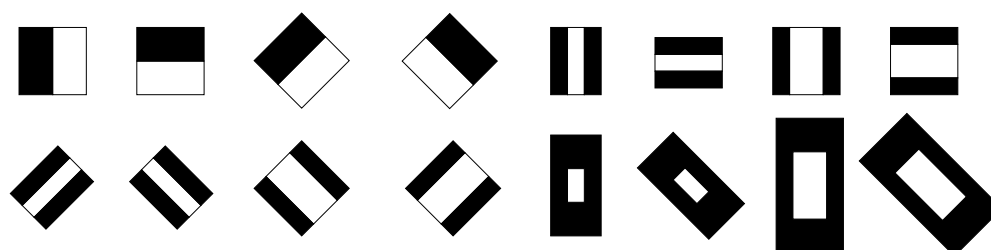
I detta arbete har olika autonoma funktioner använts såsom detektion, målföljning, återigenkänning och sensorstyrning. I detta avsnitt följer en översiktlig beskrivning av dessa funktioner.

2.1 Detektion

Automatisk måldetektion kan, åtminstone i vissa fall, på sikt tänkas ersätta människan i spaningssituationer då så stora datamängder ska avsökas att det inte är möjligt att ägna mer än något fåtal sekunder åt varje enskild bild. Datorn kan i sådana situationer hjälpa operatören genom att peka ut områden av potentiellt intresse, vilket sparar tid. Huruvida tidsbesparingen är relevant beror givetvis på hur mycket data som ska bearbetas och hur mycket tid som står till förfogande. En möjlig användning av datorbaserad detektion skulle således kunna vara att ange i vilka bilder (väl synliga) föremål och egenskaper av ett visst slag förekommer, t.ex. militära fordon, hjulspår, byggnader, vägar eller föremål i rörelse.

Detektionsalgoritmen som har integrerats till MSSLab togs från början fram för att användas på främst IR-bilder [1]. Det har dock visat sig att samma algoritm kan användas även för att detektera mål i SAR-bilder.

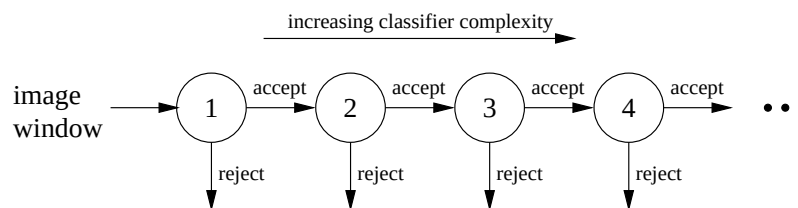
Denna detektionsalgoritm fungerar vanligtvis så att man för varje bildpunkt testat om ett lokalt fönster kring punkten innehåller ett mål. Detektionsalgoritmen genomför testet genom att applicera ett antal operatorer eller filter som känner av lokala strukturer, särdrag, i fönstret. I det aktuella fallet har filter av de typer som visas i figur 1 använts; dessa är känsliga för kanter och linjer av olika slag, samt punkter vars gråvärde avviker från omgivningen. Detektionsalgoritmen viktat sedan samman filtersvaren, och om värdet överstiger en tröskel beslutas att fönstret innehåller ett mål.



Figur 1. Filtertyper som används av den implementerade detektionsalgoritmen. Genom att variera höjd, bredd och position kan mer än 100000 olika filter genereras i fönster av storleken 20×20 bildelement.

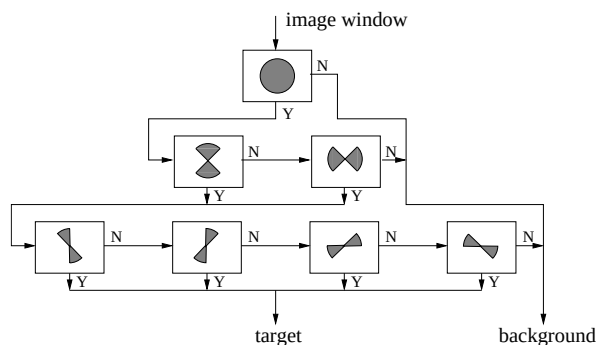
Träningen av detektionsalgoritmen består i huvudsak av att bestämma vilka filter som ska användas, samt hur filtersvaren ska viktas samman. Detektion av markmål är på grund av den svåra klottermiljön ett utomordentligt svårt problem, så för att uppnå acceptabla prestanda med avseende på falsklarms- och detektionssannolikhet krävs ett mycket stort antal filter (flera tusen). Även om svaren från filtren i figur 1 kan beräknas mycket effektivt, är det omöjligt att i

realtid applicera flera tusen filter i varje bildpunkt. En elegant lösning på detta problem är att använda en sekvens av detektorer, så som visas i figur 2. Först bearbetas bilden med en enkel detektor som har en viss förmåga att förkasta bakgrunder och en hög sannolikhet att acceptera ett verkligt mål. Om en signifikant andel (kanske hälften) av alla bildpunkter på detta sätt kan avfärdas, återstår en mindre del att bearbeta i nästa steg. Man har därför råd att använda en mer komplex detektor för att ytterligare reducera antalet bakgrundsomgivningar. Sekvensen förlängs sedan på samma sätt tills önskad falsklarms sannolikhet uppnåtts. Med denna metodik kan antalet filter som i snitt måste appliceras i varje bildpunkt reduceras med mer än 90%.



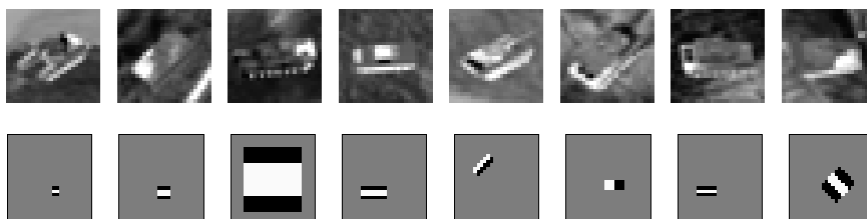
Figur 2. Sekvens av detektorer.

För att ytterligare effektivisera beräkningarna kan man även dela upp varje steg i ett antal detektorer, var och en specialiserad på ett visst vyintervall. Det är rimligt att anta att ett flygande spaningssystem med hjälp av navigeringssensorer och en terrängmodell har en god kunskap om avståndet till och orienteringen av markplanet. Markfordon undviker också i största möjliga mån branta lutningar. Sammantaget medför detta att osäkerheten om elevationsvinkeln mellan markplanet (och därmed fordonet) och spaningsplattformen är relativt liten. Azimutvinkeln, dvs. fordonets orientering i markplanet, är dock i allmänhet helt okänd. För att underlätta detektionen kan man därför träna detektorer att känna igen mål inom ett visst azimutintervall, och sedan kombinera dem enligt figur 3.



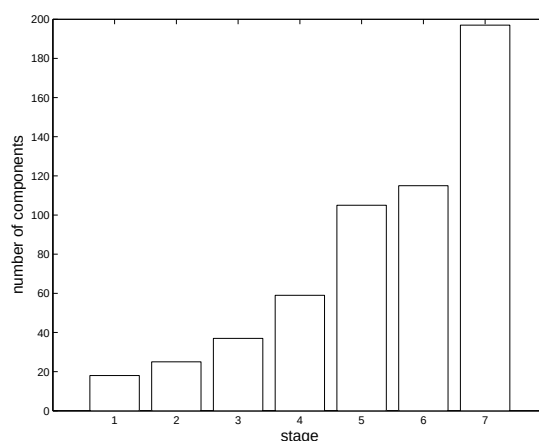
Figur 3. Hierarki av detektorer. Den sekventiella bearbetningen sker nu uppifrån. För varje steg görs en allt finare uppdelning i vyintervall för att klara det allt svårare diskrimineringsproblemet. Varje modul består i sin tur av en sekvens av detektorer.

Låt oss i mer detalj studera resultat för en modul (ett vyintervall) på den finaste nivån i figur 3. Exempel på träningsdata och några valda filter för IR-fallet visas i figur 4. Den träningsmetod som använts väljer successivt ut de filter som på bästa sätt särskiljer mål och bakgrund.



Figur 4. Exempel på träningsdata och några valda filter för att särskilja mål från bakgrund.

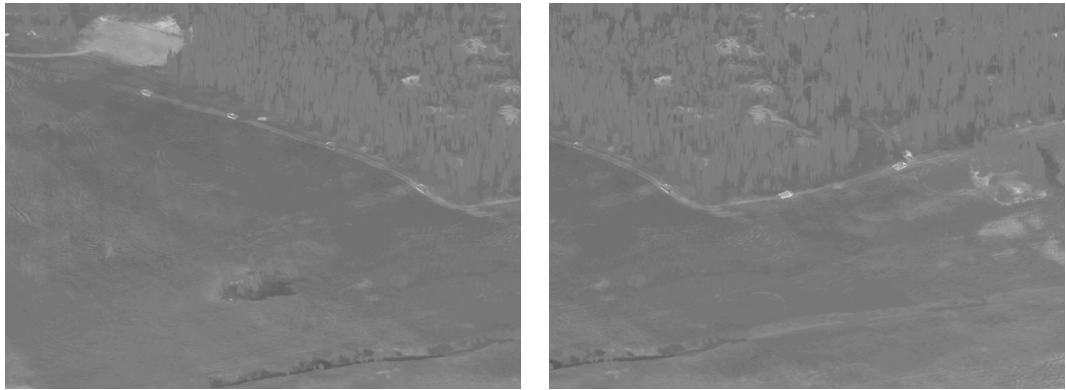
I figur 5 visas hur antalet filter i successiva steg ökar, som följd av att det blir allt svårare att särskilja återstående bakgrunder från målen.



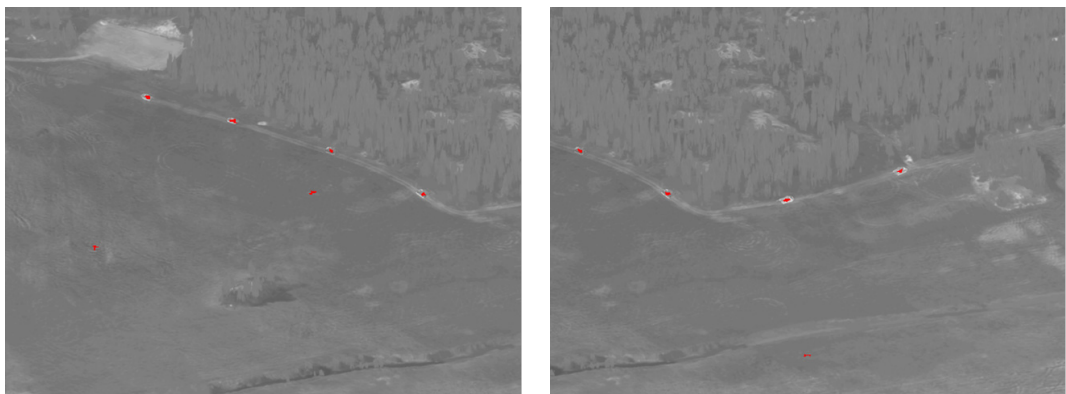
Figur 5. Antalet filter som krävs i de olika stegen i detektorsekvensen för att uppnå prestandakraven.

Algoritmen är utformad för att uppnå hög detektionssannolikhet (mer än 90 %) vid låg falsklarmsintensitet (någon enstaka indikering per bild) för signaturreducerade mål med låg kontrast mot bakgrunden. Den skall således kunna fungera även mot mål som inte kan detekteras med enkla "hot spot"-detektorer (t.ex. fordon med avstängd motor) och i komplexa klottermiljöer. För att uppnå dessa prestandakrav krävs en minsta upplösning som kan skattas till omkring 0.5 meter per pixel.

I figur 6 visas bilder ur en testsekvens med hög bildkontrast, motsvarande en solig sensommardag. Bilderna simulerar en högkvalitativ IR-sensor känslig i våglängdsbandet 8-9 μm . Sensorn har en fokalplansmatris med 640×480 pixlar. Synfältet är 5 grader i horisontell led, vilket på det aktuella målavståndet resulterar i en pixelstorlek på ca 0.4 m. Utdata från detektionsalgoritmen visas i figur 7. Detektionsprestandan är mycket god och samtliga mål detekteras. Falsklarmsintensiteten är låg med endast någon enstaka felaktig indikering. Man bör dessutom notera att falsklarmsintensiteten kan minskas ytterligare genom att applicera en målföljningsalgoritm på indikeringarna, eftersom de flesta falsklarm bara förekommer i enstaka bilder.



Figur 6. Bilder ur simulerad testsekvens.

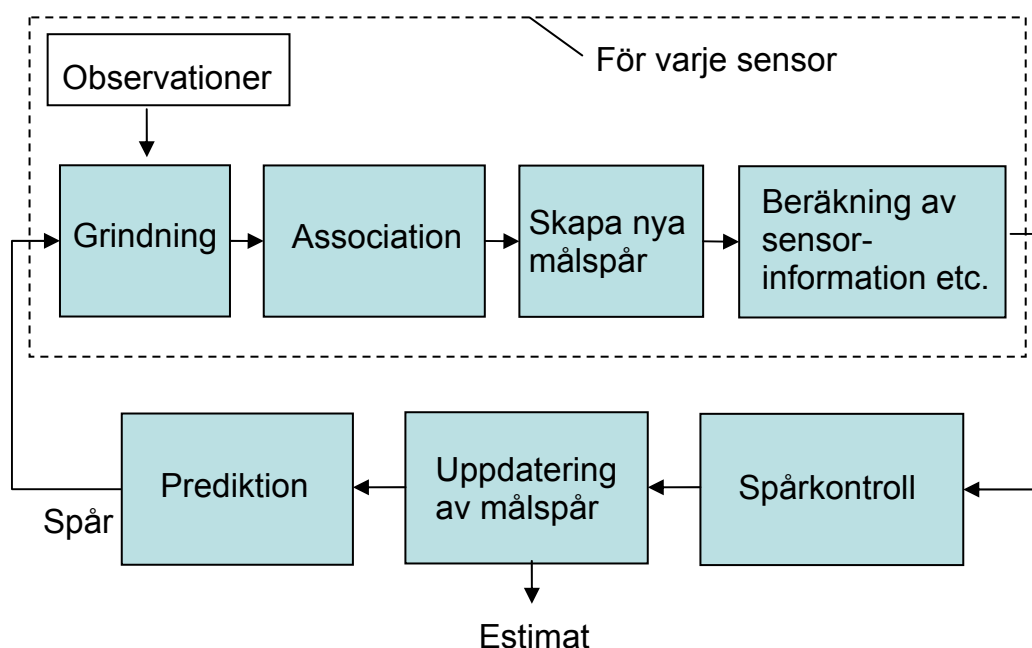


Figur 7. Indikeringar från detektionsalgoritmen för bilderna i figur 6 .

2.2 Målföljning

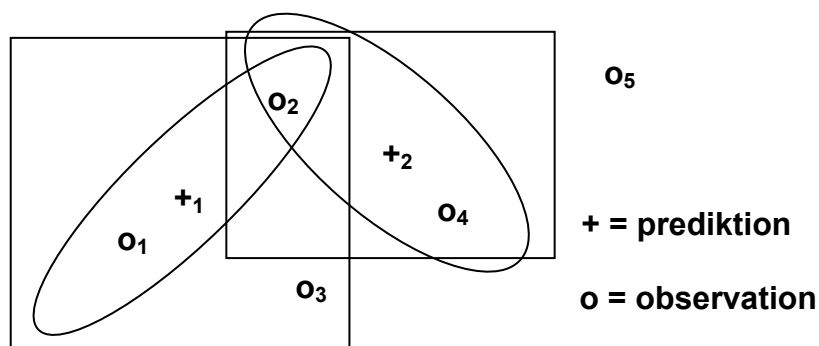
En målföljningsalgoritm har integrerats till MSSLab. Denna kallas MSMT-målföljaren (MultiSensorMultiTarget) och är utvecklad för multisensorsystem. MSMT-trackern är implementerad i C++ och grundtanken med dess design är att den ska vara allmän. Den klarar flera olika sensorer, många mål, olika rörelsemodeller och olika associationsmetoder. Nedan följer en kort beskrivning av den integrerade MSMT-målföljaren.

En målföljare har som uppgift att utifrån observationer av ett mål skapa *målspår* över tiden. Dessa *målspår* ger information om objektens rörelse och kan t.ex. användas för att estimeras målets position och hastighet. I figur 8 nedan visas en schematisk bild av MSMT-målföljaren. Denna målföljare är uppbyggd för att kunna hantera flera olika sensorer och kunna följa flera mål samtidigt.



Figur 8. Målföljningsloop för MSMT-målföljaren.

Målföljningsalgoritmen startar med att observationer från ett mål sällas med hjälp av grindarna. Detta för att underlätta för den efterföljande associationen som är beräkningstung.



Figur 9. Rektangulära och elliptiska grindar

Figur 9 visar ett exempel på hur grindar används. De fem observationerna ska associeras till prediktionerna för de två målspar. I detta fall används två grindar i sekvens, en rektangulär och en elliptisk grind. Den rektangulära grinden är minst tidskrävande och sällar snabbt bort observation nummer fem. Därefter används de bättre men mer beräkningstunga elliptiska grindarna på de återstående observationerna och observation 3 sällas bort. Efter detta återstår 3 observationer vilka ska associeras till de två målspar.

Associeringen av observationer till målspar kan ske enligt olika principer. I MSMT-målföljaren finns två associationsmetoder implementerade: *Global Nearest Neighbor* (GNN) och *Joint Probabilistic Data Association* (JPDA).

GNN (Global Nearest Neighbor) är en enkel och mycket vanlig associationsalgoritm. Den fungerar så att observationen som har det kortaste avståndet till den predikterade positionen anses vara den rätta. När det finns flera objekt i området som observationerna kan härröra från väljs de kombinationer som minimerar det totala avståndet mellan observationer och prediktioner. I exemplet ovan kommer då observation ett att associeras till målspar ett och observation fyra kommer att associeras till målspar två, eftersom dess avstånd till prediktionen är kortast.

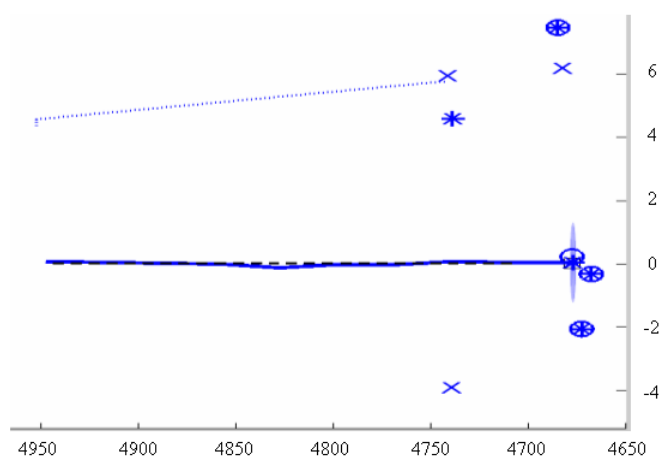
En annan mycket vanlig associationsmetod är JPDA (Joint Probabilistic Data Association). Vid uppdateringen av målspar med denna metod viktas alla associationer mellan prediktioner och observationer inom grinden samman så att associationer med observationer som ligger närmast prediktionen får störst vikt/betydelse. Därefter sammanställs estimaten baserade på dessa associationer till ett slutgiltigt estimat med hjälp av respektive vikt. I exemplet ovan kommer både observationerna ett och två att användas vid uppdateringen av målspar ett, och på samma sätt kommer observationerna två och fyra att användas vid uppdateringen av målspar två.

Många studier visar att JPDA är bättre än GNN på att bibehålla enstaka målspar i en miljö med mycket klotter. En nackdel med JPDA är dock att den har svårt att hålla isär målspar för objekt som rör sig nära varandra. Därför är det vanligt att på olika sätt kombinera GNN och JPDA för att dra nytta av båda algoritmernas styrkor.

För observationer som inte kan associeras till något målspar skapas helt nya målspar. Sensorinformation som kovariansmatriser, detektionssannolikheter etc. beräknas för respektive sensor, vilket sedan används vid den globala uppdateringen av målsparen.

För att målföljningsalgoritmen ska fungera bra behövs spårkontroll för att ta bort dåliga spår, slå ihop spår som tillhör samma objekt etc. Kvalitén på ett spår har i MSMT-målföljaren analyserats med en metod som heter SPRT (Sequential Probability Ratio Test) som beräknar ett "track score" för de olika målsparen. Målspar som associeras med en observation får då högre "track score" och målspar som inte associeras med en observation får lägre "track score". Om track score överstiger en övre tröskel sägs spåret vara konfirmerat och om track score understiger en undre tröskel tas målspar bort. Målspar som är kvar uppdateras sedan och predikteras framåt i tiden.

Nedan visas ett exempel på ett målspar. I detta exempel har målet befunnit sig på 4950 meters avstånd till sensorn och målet har sedan rört sig mot sensorn som är kopplad till målföljningsalgoritmen.

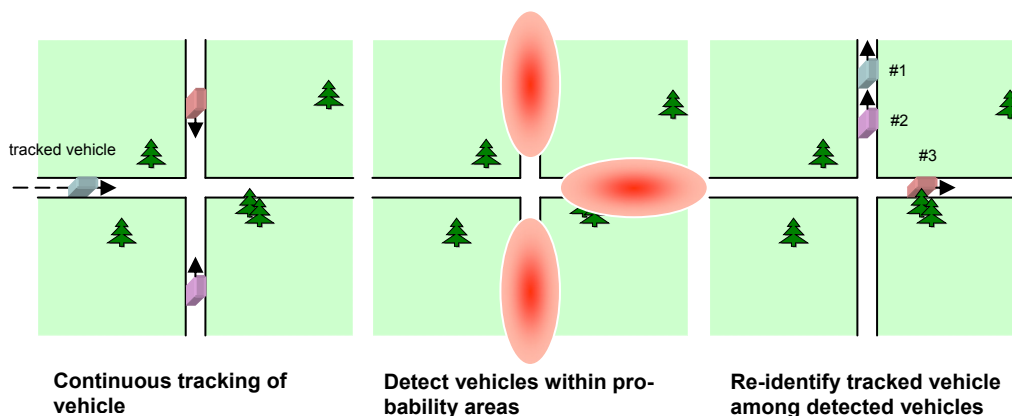


Figur 10. Ett målspar under anflygningen.

Den svarta streckade linjen visar börvärdet som en ideal målföljningsalgoritm skulle få, och den blå linjen visar målsparat. De blå stjärnorna visar spårens nuvarande estimat och de blå kryssen visar spårens prediktioner. De blå cirkelarna visar observationer som detektorerna har levererat till målföljningsalgoritmen. Den tunna streckade linjen är ett målspar som ej är bekräftat och som till slut tas bort.

2.3 Återigenkänning

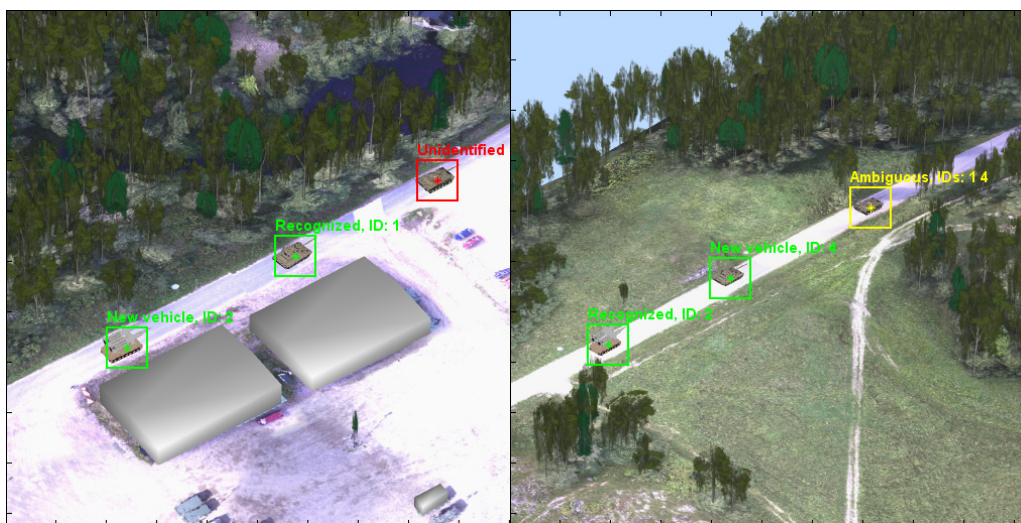
En ny funktion som integrerats i MSSLab under 2008 är återigenkänning [2]. Ett typiskt scenario för återigenkänningen visas i figur 11. Ett (rörligt) fordon på marken följs från t.ex. en UAV under ett visst tidsintervall. Under tiden fordonet är synligt extraheras robusta spatiella, temporala och spektrala särdrag från fordonet. För att lösa andra spaningsuppgifter behöver UAV:n sedan avbryta den aktiva följningen under en tidsperiod. Avbrottet kan t.ex. bero på skylning eller andra högprioriterade uppgifter. UAV:n försöker därefter återfinna det rörliga fordonet. De särdrag som tidigare extraherats används för att med hög signifikans avgöra om det är samma fordon som tidigare följts. UAV:n fortsätter sedan att följa fordonet och dynamiskt uppdatera relevanta egenskaper.



Figur 11. Typiskt scenario för återigenkänning.

En svårighet med igenkänning av fordon är att skillnaden mellan olika objektinstanser kan vara mycket liten. För bilar kan det vara dörrhandtagets läge, form hos sidoruta etc. Återigenkänningsalgoritmen måste även kunna hantera olika belysningseffekter och att fordon är synliga från olika vinklar.

Den återigenkänningsmetod som integrerats i MSSLab bygger på Color Cooccurrence Matrix (CCM) [3]. De särdrag som används i CCM är baserade på färg och textur, och representeras som histogram. En fördel med metoden är att den ger rotationsinvarianta matchningar. CCM kan användas både på naturliga färgbilder (RGB) och IR. Metodens känslighet kan justeras genom att välja färghistogrammetts upplösning (antalet kvantiseringnivåer). Om många nivåer används, kommer matchningarna att bli mer exakta under bra förhållanden, men samtidigt ökar känsligheten för förändrade ljusförhållanden och brus. Ett resultat av återigenkänningen visas i figur 12.



Figur 12. Återigenkänning av fordon på march.

2.4 Ruttplanering och sensorstyrning

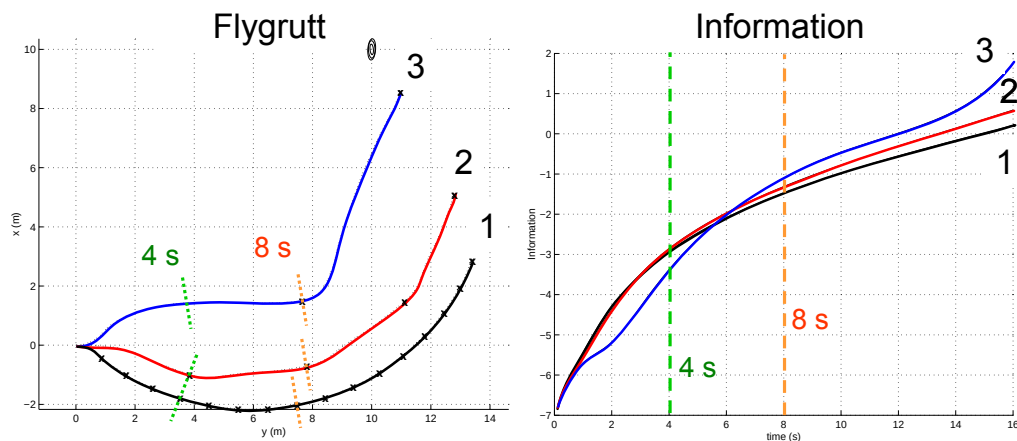
En ny funktionalitet som håller på att integreras till MSSLab är automatisk ruttplanering och sensorstyrning. Målsättningen med detta arbete är att rutt och pekriktningen för sensorn ska väljas automatiskt för att så mycket relevant sensordata som möjligt ska kunna samlas in. Vad som är relevant sensordata är beroende av målet för uppdraget och inkluderar t.ex. upptäcktssannolikheter samt prestandamått för målpositioner och målidentiteter. Dessutom bör även hänsyn tas till restriktioner som kommer ur hot- och säkerhetsaspekter.

Sammantaget kan planeringsproblemet formuleras som ett optimeringsproblem med bivillkor. Tyvärr medför dock detta angreppssätt en rad problem. För det första kan det vara svårt att kvantifiera vad som är relevant information och hur man sammanväger konkurrerande önskemål från olika intressenter. Även om man lyckas formulera en relevant målfunktion så är denna funktion i praktiken alltid icke-linjär och icke-konvex, och förmodligen även stokastisk, vilket generellt innebär att det är extremt svårt och beräkningskrävande att hitta optimum med de numeriska metoder vi är hänvisade till.

Ett sätt att hålla nere komplexiteten hos planeraren är att skapa en planeringshierarki där problemet delas upp i delproblem som vart och ett är mer hanterbart. Man måste dock vara försiktig hur denna hierarki definieras. Om detta görs på ett olämpligt sätt finns risk att bara dåliga planer genereras. Här följer ett exempel på en uppdelning av en planering i tre olika moder. Första moden innebär att man på uppmaning från t.ex. återigenkänningen eller målföljningen följer ett givet mål som befinner sig inom en (visuell) sensors vy samtidigt som UAV:n flyger på ett sätt som maximerar inmätningens prestanda. Den andra moden styr UAV:n och sensorn så att målsåret för kända mål uppdateras tillräckligt ofta för att de inte ska tappas bort. Slutligen den tredje moden är ren spaning eller, med andra ord, yttäckning för att leta efter nya intressanta objekt.

Första moden är baserad på informationsteori och är inspirerat av forskning inom optimal styrning för målföljning med passiva sensorer [4]. Antag att fordonets position är p och att vår kunskap om positionen representeras av storheten $\tilde{p}$. Eftersom vi inte är helt säkra på positionen p betraktar vi $\tilde{p}$ som en stokastisk variabel – säg för enkelhets skull en normalfördelad variabel med medelvärde m och kovariansmatris C . Mätningar vi gör ska syfta till att reducera osäkerheten så mycket det går, eller med andra ord, maximera informationen om målet. Frågan är dock hur vi kvantifierar information? I detta fall kan informationsinnehållet helt enkelt definieras som storleken av inversen på variansen $\hat{P}^{-1}$, vilket ofta kallas för Fisher-information. En fördel med detta informationsmått är att det har additiva egenskaper – det är lätt att lägga till information som kommer från nya observationer. Denna praktiska egenskap utnyttjas flitigt i de informationsfilter som är vanliga i datafusionssammanhang.

Förväntad informationsvinst kan användas vid planering av spaning, övervakning och utforskning. Om ett problem formuleras så att intressanta egenskaper representeras som tillståndsvariabler, så är en lämplig strategi att maximera informationen, eller ekvivalent, minimera osäkerheten hos variablerna. Antag att en UAV med en passiv optisk sensor ska mäta in positionen hos ett objekt på marken på kortast möjliga tid – helt i analogi med exemplet ovan. Problemet är att välja lämpliga styrsignaler så att detta krav uppfylls. Notera att en passiv sensor bara har förmåga att mäta en relativ vinkel till objektet, den har ingen möjlighet att direkt mäta avståndet. Uppgiften kan formuleras som ett optimeringsproblem där man söker en styrsignalsekvens som maximerar (minimerar) en viss målfunktion. Målfunktionen kan exempelvis vara determinanten av objektets informationsmatris $Y = \hat{P}^{-1}$ som alltså är ett mått på tillförlitligheten eller informationsinnehållet i skattningen.



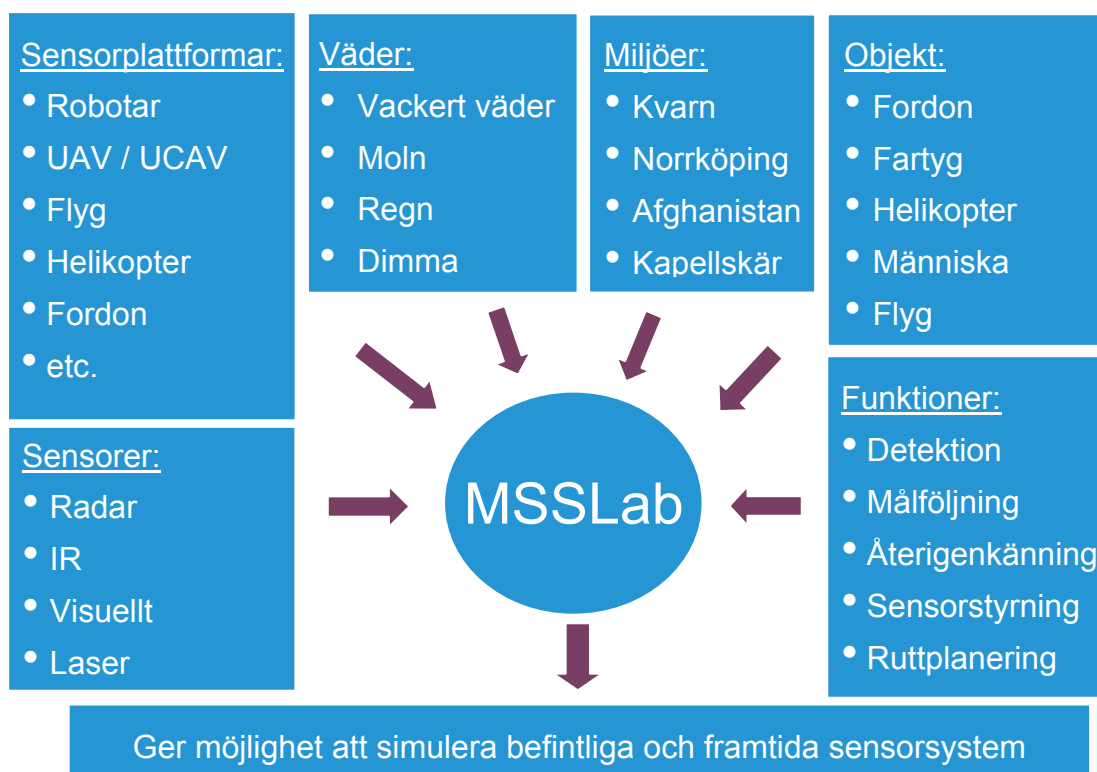
Figur 13. Lokalisering av objekt med passiv sensor i fallet med en rörelsefrihet i två dimensioner. Till Vänster: Objektet är stationärt i (10, 10), farkosten startar i (0,0). Tre olika fall med olika lång planeringshorisont visas, omplaneringar indikeras med kryss. Till höger: Informationsvärdet för de tre fallen som funktion av tiden.

Planeringen har en viss planeringshorisont och det är i regel informationsvärdet vid denna horisont som ska maximeras. Olika tidshorisonter kan ge vitt skilda resultat. I figur 13 visas planeringsresultatet för tre olika planeringshorisonter i det ovan beskrivna inmätningsexemplet. Till höger visas informationsvärdet för objektet i de tre fallen. En kort horisont (1) medför att farkosten inledningsvis färdas i princip vinkelrätt relativt riktningen till objektet. Detta beror på att den passiva sensorn enbart kan mäta vinkeln till objektet. Med en lång horisont går farkosten mer rakt på för att vika av mot slutet. Jämför man informationen ser man att resultatet med den korta horisonten (1) har mest information till en början, men passerar av den mellanlånga horisonten (2) vid $t = 3$, och av den långa horisonten (3) vid $t = 6$. Med lång horisont (3) har man alltså råd att inhämta lite information inledningsvis om man vet att man kommer få mycket information på slutet.

3 Simuleringsmiljön MSSLab

De autonoma funktionerna har integrerats till simuleringsmiljön MSSLab - MultiSensorSimuleringsLab[5]. Detta har gjorts i samarbete med projekten ARCUS och Simuleringsbaserade metoder för sensorvärdering. I detta avsnitt följer en beskrivning av några av de simuleringsmodeller som finns i MSSLab och som har använts i detta arbete.

MSSLab är en simuleringsmiljö som utvecklas vid FOI Informationssystem. I denna simuleringsmiljö kan avancerade sensorsystem simuleras i olika miljöer och vid olika väderförhållanden och tidpunkter. Simuleringsmiljön kan bidra med information om sensorprestanda för alla nivåer från teknisk till taktisk värdering. Figur 14 visar olika beräkningsenheter som kan integreras vid en simulering med MSSLab.



Figur 14. Sammanställning av delsystem som kan integreras vid en simulering med MSSLab.

Simuleringsmiljön möjliggör simulering av dynamiska scenarier med rörliga plattformar, sensorbärare, målobjekt etc. Högkvalitativa sensorsimuleringar som bygger på fysikaliska modeller kan utföras för sensorerna IR, visuellt, laser och radar. Dessa sensorsystem kan simuleras både enskilt, som samverkande sensorsystem och som multisensorsystem.

3.1 Terrängmodellen

Terrängmodellen som har använts i detta projekt är Kvarnmodellen. Denna modell representerar en modell av Prästtomta skjutfält vid Kvarn som ligger ca 3 mil nordväst om Linköping. Detta område har mätts upp med ett flygburet

lasersystem och från dessa data har terrängmodellen skapats. Kvarnmodellen innehåller en högupplöst markyta, automatiskt rekonstruerade byggnader och modeller av enskilda träd inplacerade med korrekta dimensioner på korrekta positioner. Den rumsliga upplösningen i modellen är ca 5-10 punkter/m² med en koordinatnoggrannhet på 0.1 m (i rikets nät). Detta kan jämföras med Lantmäteriverkets standardprodukt med 50 meter mellan höjdvärdena.



Figur 15. Bilden visar en del av Kvarnmodellen med uppmätt visuell textur.

Högupplöst visuellt och IR-data finns även insamlat över det laseruppmätta området. Dessa data har sedan använts som textur på den tredimensionella omgivningsmodellen. Detta ger en omvärldsmodell som ger en bra bild av hur terrängen såg ut vid det aktuella insamlingstillfället, vilket visas i figur 15 och figur 16. Terrängmodellerna med IR respektive visuell textur är båda geografiskt och geometriskt korrelerade.



Figur 16. Bilden visar en del av Kvarnmodellen med uppmätt IR-textur.

3.2 Objektmodeller

Vid värdering av sensorsystem med simuleringsbaserade metoder är det mycket viktigt att modellerna för såväl bakgrund som objekt är av hög kvalitet. På FOI finns mycket god kunskap om beräkning av signaturen hos olika objekt. Några exempel på tillgängliga objekt visas i figur 17.



Figur 17. Exempel på några tillgängliga objektmodeller

Fordonen som har använts i simuleringarna visas i figur 18. Fordonen visas med visuell och IR-textur.



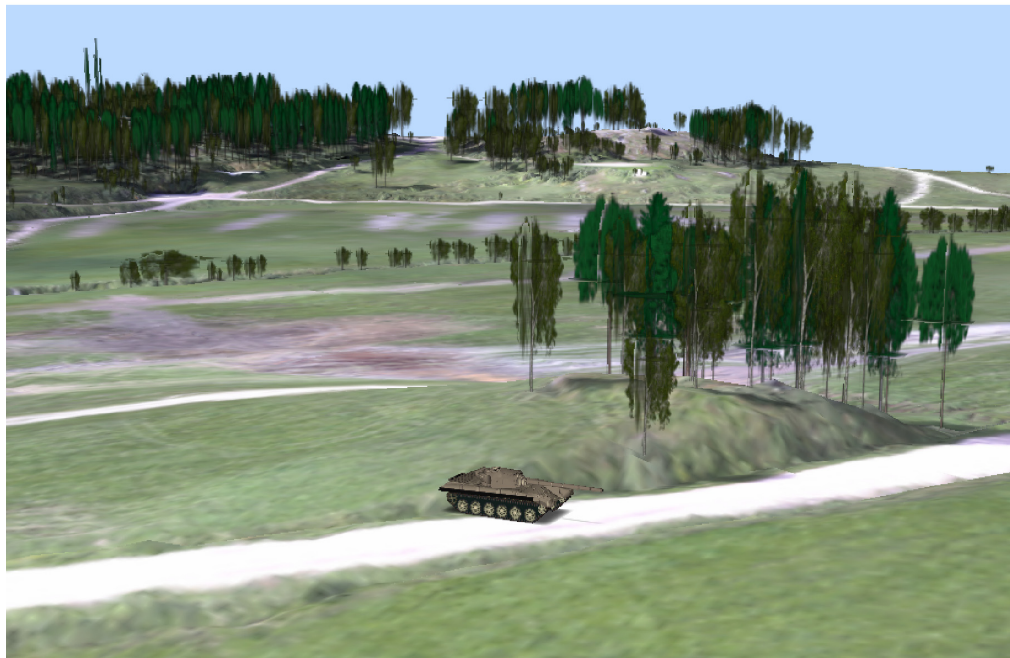
Figur 18. Använda objekt i simuleringen. De egna fordonen i kolonnen består av BTR80. De fiendliga fordonen består av T72 och de neutrala fordonen består av Humvee.

3.3 Scensimulering

För simulering av sensordata har SceneServer[6] använts. SceneServer är ett program som utvecklats av FOI för realtidsvisualisering av syntetiska omgivningar. Eftersom programmet är utvecklat av FOI har vi full insikt i koden,

vilket gör det möjligt att skräddarsy applikationer och anpassningar. SceneServer är baserat på OpenSceneGraph [7] som är ett scenograf-API för OpenGL.

SceneServer är inte baserat på fysikalisk modellering utan scenen representeras av 3D-geometrier med påklistrade texturbilder. På terrängmodellen som beskrevs i avsnitt 3.1 har texturbilder från visuellt data respektive IR-data klistrats på, vilka insamlades samtidigt med laserskanningen av terrängen. Detta ger en mycket god representation av hur terrängen såg ut vid data-insamlingstillfället, men ger endast begränsade möjligheter att variera scenen för olika väder och tidpunkter. Vid kvalificerade simuleringar av visuella och IR-sensorer används därför helst CameoSim i kombination med RadThermIR.



Figur 19. Visuellt bild genererad med SceneServer.

4 Autonoma funktioner för intelligenta operatörsstöd

4.1 Scenario

Ett scenario har använts för att studera hur ett operatörsstöd kan förbättras genom att utnyttja autonoma funktioner. Scenariot som använts består av ett UAV-system som är utrustat med IR/EO-sensorer. UAV-systemets uppgift var att autonomt följa den egna fordonskolonnen för att spana efter hot och vid behov förse kolonnen med information så att den kan omplanera sin rutt. För att UAV-systemet ska kunna göra detta autonomt behöver den kunna upptäcka och följa fordonen i terräng. Den bör även kunna inhämta sensordata för att göra det möjligt att identifiera fordonen, vilket är viktig information vid beslut om eventuell bekämpning. UAV:n bör även kunna göra en bra inmätning av målens koordinater.

Simuleringarna har utförts i simuleringsmiljön MSSLab. Fordonskolonnen kommer i scenariot att möta både fientliga och neutrala fordon. I figur 20 visas fordonskolonnens rörelse med blått, de neutrala fordonens rörelse med grönt och de fientliga fordonens rörelse med rött.



Figur 20. Figuren till vänster visar fordonsrörelserna under första halvan av scenariot och figuren till höger visar andra halvan av scenariot. Blå markeringarna visar fordonskolonnens väg, röda markeringar visar de fientliga fordonens väg och de gröna markeringarna visar de neutrala fordonens väg.

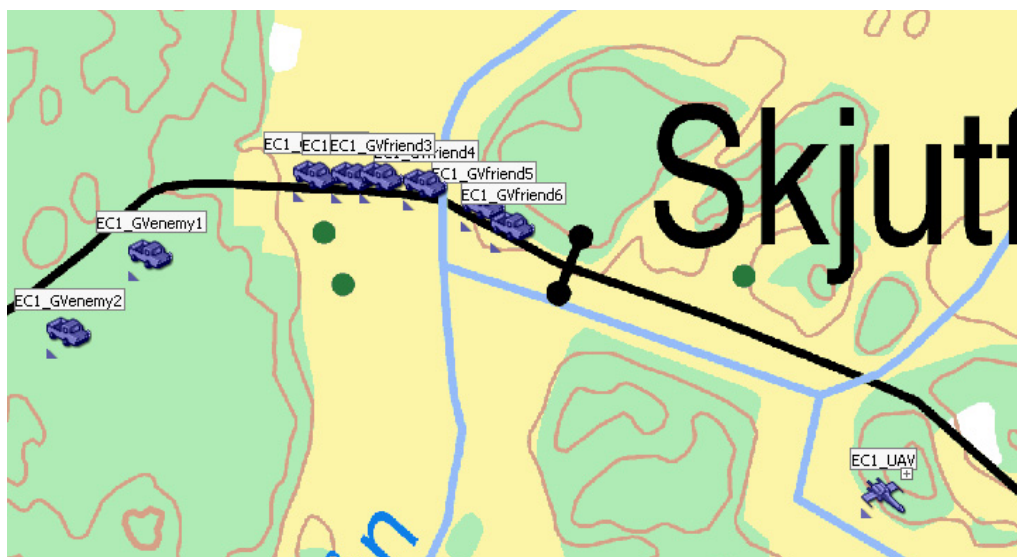
Flygbanan för UAV:n visas i figur 21. De röda markeringarna visar dess flygbana under första halvan av scenariot och de gula markeringarna visar dess flygbana under den andra halvan.



Figur 21. UAV-systemets flygbana.

4.2 Användning av autonoma funktioner för intelligenta operatörsstöd

I detta avsnitt följer en beskrivning av simuleringen som har gjorts för att visa hur de autonoma funktionerna kan bidra till ett intelligent operatörsstöd. I detta avsnitt visas de olika autonoma funktionerna vid ett visst tillfälle i scenariot. I figur 22 visas en kartbild med UAV:n, fordonskolonnen och två av de fiendliga fordonen vid det aktuella tillfället.



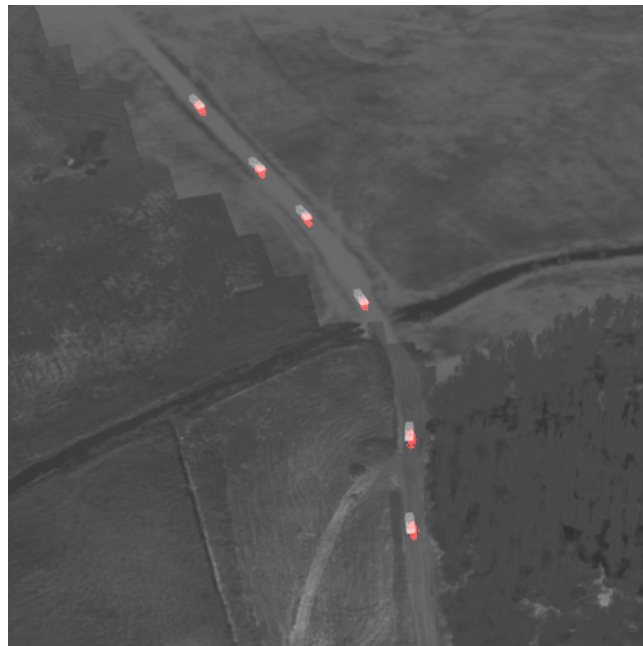
Figur 22. Bild från det aktuella scenariot.

I figur 23 visas en IR-bild från den simulerade UAV:n vid det aktuella tillfället. I bilden syns målen mycket tydligt då de rör sig på en väg över ett fält. Faktorer som rörelseoskärpa, döda pixlar, varma stenar etc. gör dock att det ofta är svårare att detektera mål i en IR-bild.



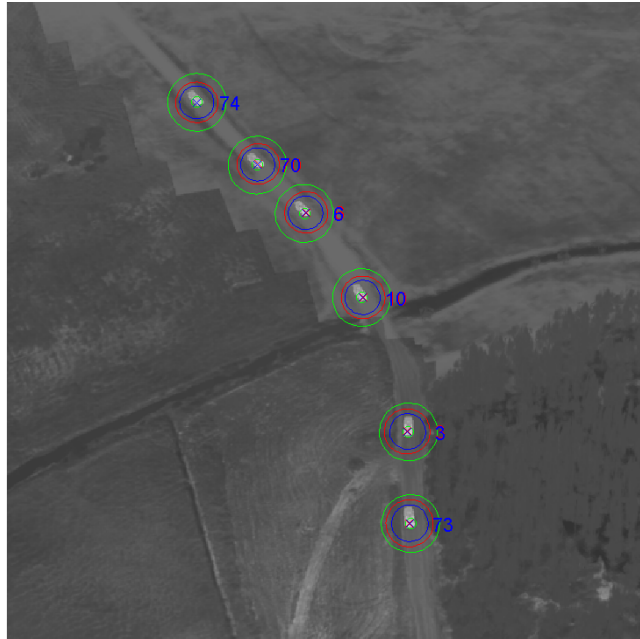
Figur 23. IR-data från scenariot.

En detektionsalgoritm har sedan använts på IR-bilderna. I figur 24 visas en IR-bild med överlagrade måldetektioner. De röda detektionerna bidrar till att för enkla för en operatör att hitta målen i bilden.



Figur 24. IR-data med överlagrade detektioner.

Detektionerna skickas sedan vidare till en målföljningsalgoritm som skapar målspar över tiden, vilket visas i figur 25. Från målföljaren kan målens positioner och hastigheter sedan estimeras.



Figur 25. Målföljning. Mätningar är markerat med grönt, uppdaterat målspar är markerat med blått och predikterat målspar är markerat med rött.

En återigenkänningsalgoritm har även använts i simuleringen, vilket visas i figur 26. Alla fordonen i konvojen består av BTR80 och återigenkänningsalgoritmen har identifierat alla objekt som ID:1, vilket är identitetsnumret för BTR80.



Figur 26. Återigenkänning.

I simuleringen har även ett presentationsverktyg som heter SpatialDisplay använts. I detta program har fordonens positioner överlagrats på en karta, vilket visas med de trekantiga symbolerna. Positionen för UAV:n visas även i programmet.



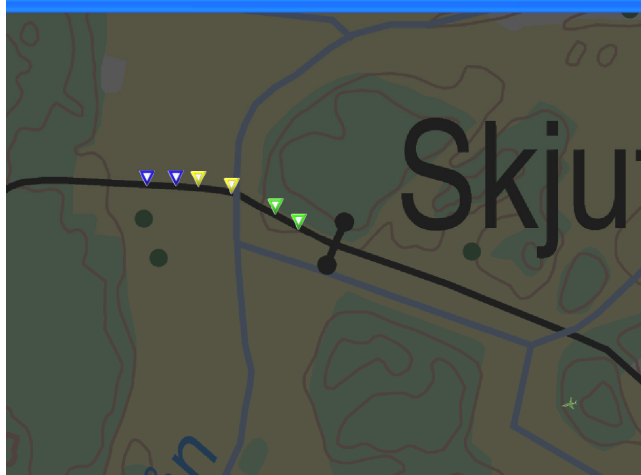
Figur 27. SpatialDisplay visar fordonskolonnens och UAV:ns position.

Om man klickar på en av de trekantiga symbolerna i SpatialDisplay skickas en interaktion till sensorstyrningen som då zoomar in med IR-sensorn mot den aktuella positionen. Ett enkelt användargränssnitt har tagits fram, vilket visas i figur 28. Genom att jämföra de inzoomade IR-bilderna med användargränssnittet underlättas en manuell bestämning av identiteten på fordonen. Användargränssnittet är uppbyggt så att i den nedre delen av gränssnittet visas några vanligt förekommande målobjekt. Genom att klicka på någon av dessa bilder visas det markerade målobjektet från fyra olika vinklar. När operatören har hittat rätt mål trycker den OK och om den inte kan bestämma identiteten på objektet så trycker den Unknown. Referensdata kommer i detta fall från simuleringar, men även uppmätt data kan användas. Referensdata som presenteras bör i ett operativt system ta hänsyn till aktuellt väder vilket ofta påverkar signalaturen på objekten. Användargränssnittet kan även användas vid träning då det krävs mycket erfarenhet att kunna känna igen ett objekt i t.ex. en IR-bild eller en SAR-bild eftersom människan inte är van att se objekten i denna signaldomän.



Figur 28. Användargränssnitt för att underlätta manuell bestämning av fordonsidentitet.

Fordonen i databasen är associerade till klasserna: vän, fiende, neutral och okänd. Klasserna är representerade med olika färger så att blått betyder vän, rött betyder fiende, grönt betyder neutral och gult betyder okänd. Då man markerat ett av de trekantiga symbolerna i SpatialDisplay och identifierat fordonet med användargränssnittet kommer färgen på den trekantiga symbolen att ändras. I figur visas ett exempel på detta.

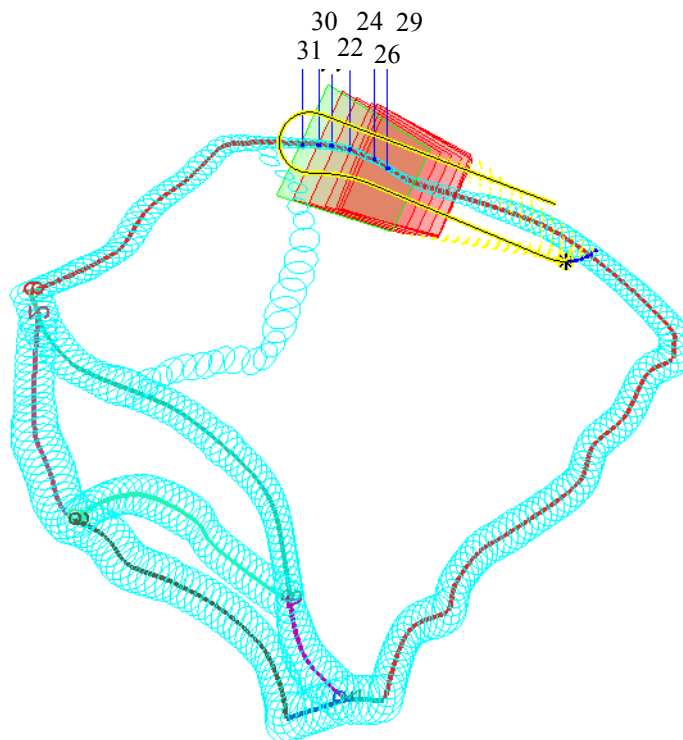


Figur 29. Bild från SpatialDisplay där symbolerna för de olika fordonen har markerats med olika färger för att visa om de är vän, fiende, neutral eller okänd.

Autonom sensorstyrning har även använts i simuleringen. Sensorstyrningen kan operera i olika moder och byta mellan dessa. Moderna som är implementerade idag är: sökning efter nya mål, styrning av sensorn så att befintliga målspår uppdateras tillräckligt ofta för att de inte ska tappas bort och styrning av sensorn för att maximera inmätningens prestanda för ett givet mål.

I figur 30 visas en planeringssekvens från den autonoma sensorstyrningen. Det gröna området visar det aktuella synfältet och de röda områdena visar de planerade synfälten. I den aktuella simuleringen gjordes en planering 15 tidssteg framåt och denna plan genomfördes innan en ny plan togs fram.

Området som ska övervakas representeras av ett antal punkter och hur väl en viss punkt är avspanad representeras med en osäkerhetsellips. I figuren är dessa punkter i huvudsak utplacerade längs med vägarna och osäkerhetsellipserna är cyan-färgade. Ju mindre ellipsen är desto bättre är denna del av vägen avspanad.



Figur 30. Autonom sensorstyrning. Det gröna området visar det aktuella synfältet och de sex röda områdena visar planerade synfält. Den gul-svarta banan markerar UAV:ns rutt.

4.3 Resultat

Resultat från simuleringarna visar att de autonoma funktionerna ger betydelsefull information till ett operatörsstöd. När enstaka bilder ska analyseras, vilket visas i avsnitt 4.2, är vinsten med autonoma funktioner inte så stor. När ett längre scenario används sjunker dock kapaciteten hos en operatör relativt fort, vid spaning efter mål. De autonoma funktionerna ger då ett betydelsefullt stöd. Projektets omfattning har inte gjort det möjligt att ta fram kvalitativa mått på hur mycket de olika autonoma funktionerna bidrar till ett intelligent operatörsstöd. Scenariot som använts var 18 minuter långt och detta scenario är relativt begränsat till både yta och antal fordon. I mer komplexa scenarier som t.ex. ett urbant scenario med många fordon, som dessutom kan försvinna tillfälligt ur synfältet t.ex. på grund av skylning, är det dock troligt att fördelarna med att använda autonoma funktioner blir ännu tydligare.

5 Slutsatser och fortsatt arbete

Syftet med projektet Autonoma funktioner var att studera hur ett operatörsstöd kan förbättras genom att utnyttja sensordataanalys och autonoma funktioner. I projektet har följande autonoma funktioner använts: detektion, följning, återigenkänning och sensorstyrning.

För att kunna studera möjligheterna med de olika autonoma funktionerna har dessa integrerats till simuleringsmiljön MSSLab. En demonstration har genomförts inom projektet som visade hur de autonoma funktionerna kan bidra till ett operatörsstöd. Scenariot som användes bestod av en UAV som flög före en fordonskolonn för att spana efter hot, så att kolonnen vid behov kunde omplanera sin rutt utifrån informationen från UAV:n.

Resultaten från simuleringarna visar att de autonoma funktionerna ger bra stöd vid spaning efter fientliga fordon. Framför allt då ett längre scenario används märktes det att spaningskapaciteten avtog, och då ger de autonoma funktionerna ett betydelsefullt stöd.

5.1 Fortsatt arbete

Projektet Autonoma funktioner har nu avslutats, men det finns fortfarande många frågeställningar kvar att studera inom området autonoma funktioner för operatörsstöd. Utvecklingen av autonoma funktioner går mycket fort och den ökade datorkapaciteten möjliggör allt mer avancerade algoritmer. Detta kommer att medföra att framtida operatörer kan få ett allt bättre stöd.

Några kvalitativa mått på hur betydelsefulla de använda autonoma funktionerna har varit har inte rymts inom detta projekt. I en fortsättning bör det även ingå en behovsanalys för att undersöka om några andra autonoma funktioner behövs såsom automatisk måligenkänning, ruttplanering etc.

I en fortsättning bör mer komplexa scenarier studeras såsom urbana scenarier där ännu högre krav ställs på ett operatörsstöd. I scenariot som användes fanns det en UAV, men detta kan enkelt utökas till flera UAV:er om man vill undersöka vilket stöd som behövs för att en operatör ska kunna sköta flera plattformar samtidigt.

6 Referenser

- [1] Karlholm J., "Implementation and evaluation of a method for detection of ground targets in aerial EO/IR imagery", Linköping, FOI, Scientific report, FOI-R--1267--SE, ISSN 1650-1942, Juni 2004.
- [2] Karlholm J., Rydell J., Stenborg K-G., Ulvklo M., Nygårds J., Hemström F., "Reidentification of ground targets in EO/IR-imagery", FOI-R--2586--SE, ISSN 1650-1942, October 2008.
- [3] Chang P. och Krum J., "Object recognition with color cooccurrence histograms", IEEE Conference on Computer Vision and Pattern Recognition, 1999.
- [4] Skoglar P., Nygårds J., Björström R., Ögren P., Hamberg J., Hermansson P., Ulvklo M., "Path and Sensor Framework Applicable to UAV Surveillance with EO/IR sensors", FOI-R--1741--SE, september 2005.
- [5] Näsström F., Karlsson M., Carlsson L., Chevalier T., Follo P., Hedström J., Karlsson N., Persson A., Sume A., "Simuleringsbaserade metoder för sensorvärdering", Underlagsrapport, FOI-R--2629, December 2008.
- [6] Bennet F. och Fenelius S., "SceneServer - a 3D software assisting developers of computer vision algorithms", FOI. Linköping, Technical report, FOI-R--0831--SE, March 2003.
- [7] Internet: <http://www.openscenegraph.org>, besökt 2007-12-16.