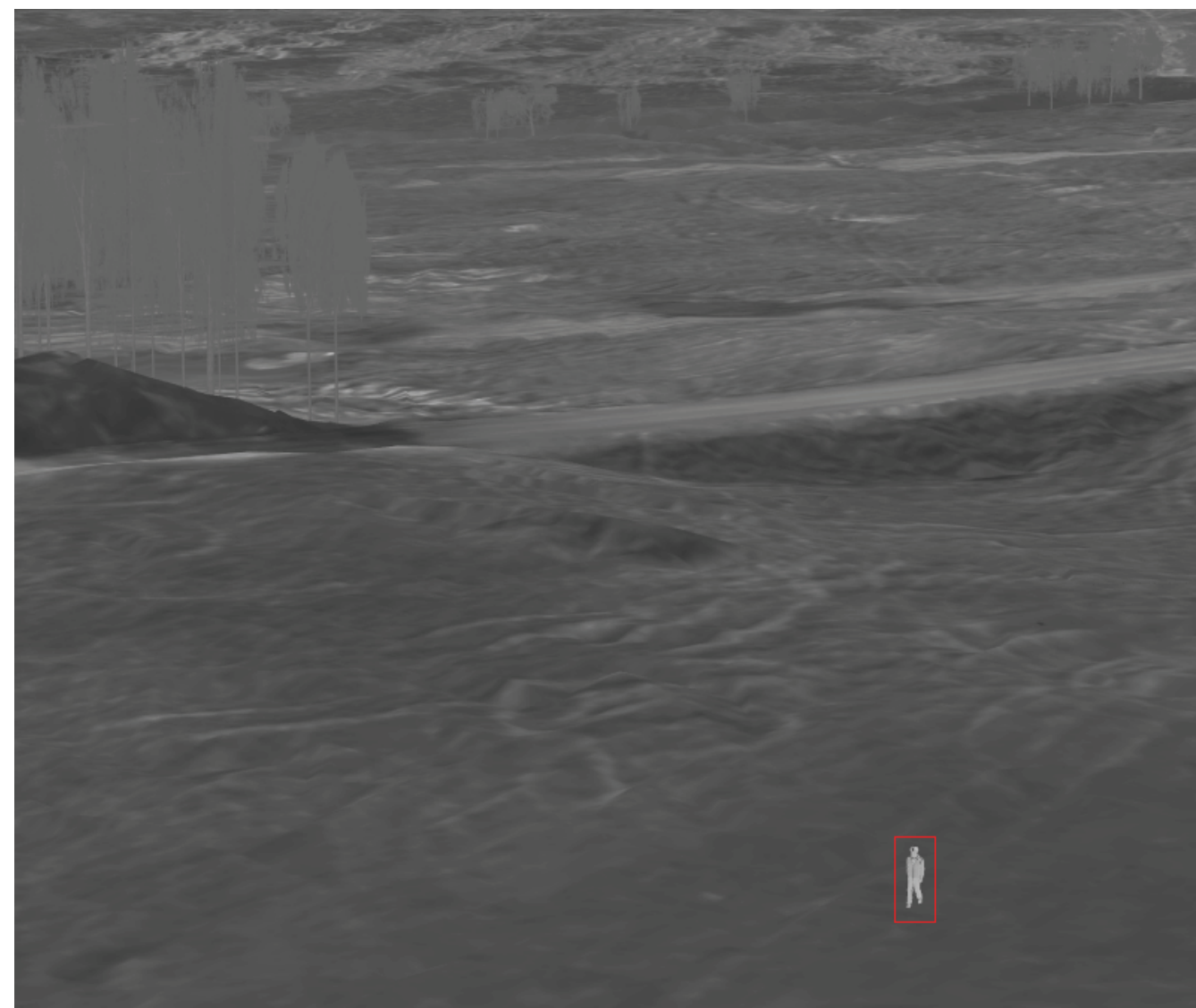


FREDRIK NÄSSTRÖM, JONAS ALLVAR, JIMMY BERGGREN,
TOMAS CHEVALIER, ROBERT FORSGREN, HANS HABBERSTAD,
JOHAN HEDSTRÖM, FREDRIK HEMSTRÖM, GUSTAF HENDEBY,
MIKAEL KARLSSON, ANDREAS PERSSON



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, Jonas Allvar,
Jimmy Berggren, Tomas Chevalier,
Robert Forsgren, Hans Habberstad,
Johan Hedström, Fredrik Hemström,
Gustaf Hendeby, Mikael Karlsson,
Andreas Persson

Simuleringsbaserade metoder för sensorvärdering

Årsrapport 2011

Titel	Simuleringsbaserade metoder för sensorvärdering – Årsrapport 2011
Title	Simulation based sensor system assessment – Annual report 2011
Rapportnr/Report no	FOI-R--3357--SE
Rapporttyp/ Report Type	Användarrapport
Sidor/Pages	41 p
Månad/Month	December
Utgivningsår/Year	2011
ISSN	ISSN 1650-1942
Kund/Customer	FM
Projektnr/Project no	E53331
Godkänd av/Approved by	Lars Bohman

FOI, Totalförsvarets Forskningsinstitut

Avdelningen för Informationssystem
Box 1165
581 11 Linköping

FOI, Swedish Defence Research Agency

Information Systems
Box 1165
SE-581 11 Linköping

Sammanfattning

Projektet SIMSENS (Simuleringsbaserade metoder för sensorvärdering) har i uppdrag att utveckla en gemensam simuleringsmiljö för värdering av sensorsystem i olika miljöer, väderförhållanden och tidpunkter. Simuleringsverktyget som vidareutvecklas för detta är MSSLab (MultiSensorSimuleringsLab). I SIMSENS-projektet har olika simuleringsmodeller för terräng, objekt och sensorer utvecklats för att göra detta möjligt. I denna rapport beskrivs arbetet som bedrivits i projektet under 2011.

För att kunna göra noggranna simuleringar av sensorsystem i olika miljöer krävs terrängmodeller med hög kvalitet. I MSSLab kan sensorsystem nu simuleras i terrängmodellerna: Kvarn, Norrköping och Mellanöstern. Ett mycket viktigt arbete har varit att ta fram en automatisk metodik för materialklassificering av terrängmodeller. Den framtagna metodiken har i år använts för att materialklassificera de tre omvärldsmodellerna.

För noggranna simuleringar av sensorsystem är det även mycket viktigt att objektmodellerna är av hög kvalitet. Ett 3D-modellbibliotek med olika objekt och människor har vidareutvecklats för detta. För att göra det möjligt att simulera sensordata med animerade människor har HLAS (High Level Animation System) vidareutvecklats med nya beteenden för de animerade människorna.

Ett omfattande arbete har varit att vidareutveckla de olika sensormodellerna i MSSLab. Under året har sensormodeller för akustik, IR och laser vidareutvecklats inom projektet. Ett scenario har simulerats med de vidareutvecklade simuleringsmodellerna för att studera sensorprestanda ur ett systemperspektiv. Syftet med det simulerade scenariot är att studera möjligheten att automatiskt upptäcka människor från ett UAV-system. Prestanda för detektion och målföljning har studerats för detta scenario. Ljudutbredningen från ett UAV-system har även simulerats, för att studera det akustiska upptäcktsavståndet.

Nyckelord: simulering, modellering, värdering, akustik, IR, laser, radar

Summary

The task for the SIMSENS' project (Simulation based sensor system assessment) is to develop a modular sensor system simulation program that can simulate sensor systems in various environments, weather conditions and time periods. The simulation tool that is further developed is the MSSLab (MultiSensorSimulationLab). In the SIMSENS' project various simulation models for terrain, objects and sensor, have been improved to make this possible. This report describes the work that has been performed in the project during 2011.

In order to perform accurate simulation of sensor systems in various environments high quality models of the terrain are required. In MSSLab sensor systems can now be simulated in the terrain models: Kvarn, Norrköping and Afghanistan. A very important work has been to develop an automated methodology for automatic material classification of terrain models. The developed methodology has this year been used to classify the material of the three terrain models.

For accurate simulations of the sensor system it is also very important that the object models are of high quality. A 3D model library of various objects and people has therefore been further developed. To make it possible to simulate sensor data with animated persons, HLAS (High-Level Animation System) has been further developed with new behaviors for animated people.

Extensive work has been to further develop the different sensor models in MSSLab. During this year, sensor models for acoustic, infrared and laser have been further developed within the project. A scenario has been simulated with the new simulation models to study the sensor performance on a system level. The purpose of the simulated scenario is to study the possibility of automatically detecting people from a UAV. The performance of the detector and tracker has been studied for this scenario. Sound propagation from a UAV system has also been simulated to study the acoustic detection distance.

Keywords: simulation, modeling, assessment, acoustics, IR, laser, radar

Innehållsförteckning

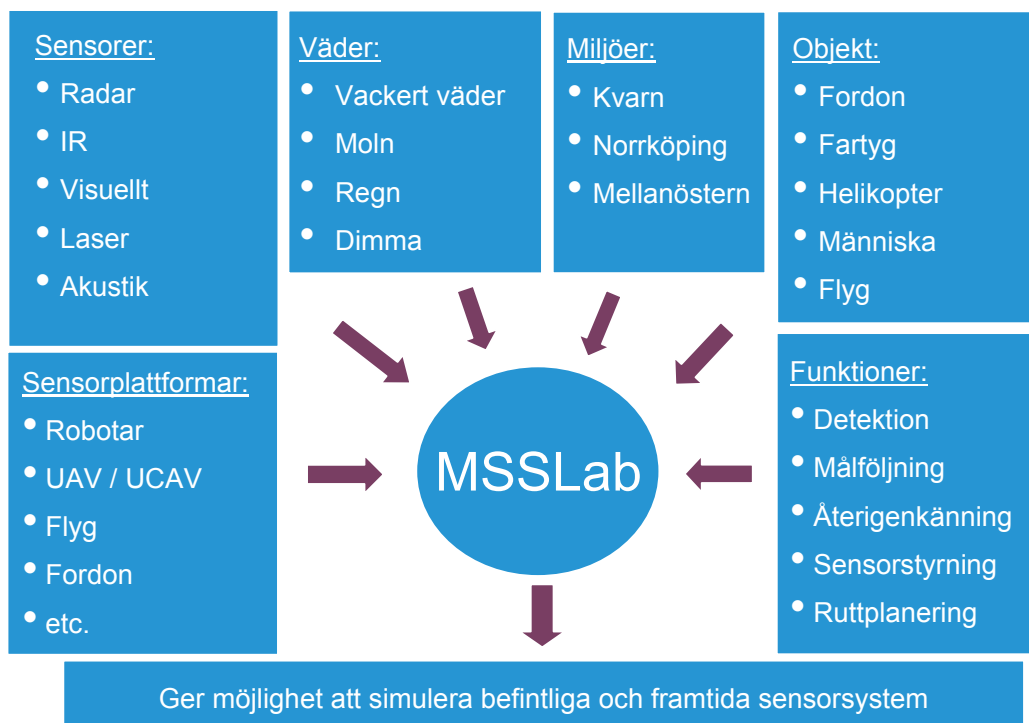
1	Inledning	7
2	Terrängmodeller	9
2.1	Norrköpingsmodellen.....	9
2.2	Kvarnmodellen.....	11
2.3	Mellanösternmodellen.....	15
3	Objektmodeller och animering av människor	18
3.1	Objektmodeller med visuell textur.....	18
3.2	Beräkning av IR-signaturer av människor.....	19
3.3	Animering av människor	20
4	Distribuerade simuleringar	24
4.1	Scenarioverktyget Netscene.....	25
4.2	MSSLab-federater	25
4.3	Batchkörningar med MSSLab.....	26
5	Simulering av sensorsystem	27
5.1	Visuella simuleringar av ett marint scenario.....	27
5.2	Simulering av ljudutbredning från UAV-system.....	28
5.3	Simulering av IR i olika miljöer.....	31
5.4	Simulering av 3D-avbildande laserradar.....	32
6	Resultat från värdering av spaning efter människor	36
6.1	Detektion	37
6.2	Målföljning	38
7	Fortsatt arbete	40
8	Referenser	41

1 Inledning

Vid värdering av ett sensorsystem krävs tillförlitlig information om sensorprestanda för den aktuella uppgiften. Spelkort eller tekniska specifikationer finns ofta för befintliga system, men information om sensorprestanda för många internationella och framtida sensorsystem, och sensorsystem i nya miljöer är ofta otillräcklig.

Ett vanligt sätt att skatta prestanda hos ett system är med hjälp av simuleringsbaserade metoder. För att få hög tillförlitlighet med simuleringsbaserade metoder krävs god kännedom om det aktuella systemet samt validerade modeller. Fördelen med simuleringsbaserade metoder är att när väl systemet är modellerat och omgivningsmodellerna finns, kan simuleringar göras för en mängd olika fall till en rimlig kostnad och på relativt kort tid.

Genom projektet "Simuleringsbaserade metoder för sensorvärdering" utvecklas en gemensam simuleringsmiljö för värdering av sensorsystem. Denna simuleringsmiljö ska även kunna bidra med information om sensorprestanda för alla nivåer från teknisk till taktisk värdering. Simuleringsverktyget som vidareutvecklas för detta är MSSLab (MultiSensorSimuleringsLab). MSSLab är en simuleringsmiljö där avancerade sensorsystem kan simuleras i olika miljöer och vid olika väderförhållanden och tidpunkter. Figur 1 visar olika beräkningseenheter som kan integreras vid en simulering med MSSLab.



Figur 1. Sammanställning av delsystem som kan integreras vid en simulering med MSSLab.

Simuleringsverktyget möjliggör simulering av dynamiska scenarier med rörliga plattformar, sensorbärare, målobjekt etc. Högkvalitativa sensorsimuleringar kan göras för IR, visuellt, laser och radar. Dessa sensorsystem kan simuleras både enskilt, som samverkande sensorsystem och som multisensorsystem. Avancerade algoritmer för ob-

jektdetektion, följning, återigenkänning, sensorstyrning och ruttplanering har även integrerats till simuleringsverktyget.

En svårighet vid simulering av sensorsystem är att sensorprestanda påverkas av många variabler. MSSLab har redan idag mycket bra funktionalitet, men mycket arbete återstår för att få hög tillförlitlighet vid värdering av sensorsystem i nya miljöer. Exempel på viktiga scenariovariabler som ofta påverkar sensorprestanda är:

- Målobjekt: konfiguration, signatur, kamouflage, skylda eller ej, mål på marsch eller stillastående etc.
- Terräng: bebyggelse, öppen, småbruten, betäckt, öken, djungel etc. Terrängen kan dessutom vara platt, kuperad eller bergig, samt ha inslag av sjöar och vattendrag.
- Vegetation: fördelningar av träd och buskar och dess årstidsförändringar.
- Väder/atmosfärstransmission: geografisk plats, årstid, dygnstid, vind, luftfuktighet, temperatur, molnighet, molnbas, olika typer av nederbörd och intensitet etc.
- Ljuszförhållanden: tid på året och dygnet, aktuellt väder och geografisk plats.
- Datainsamling: observationstid, avstånd, höjd, aspektvinkelförändring på målet under observationen, hastighet på sensorplattformen etc.
- Hot och störning mot sensor och sensorbärare.
- Handlungsregler för uppdraget (ROE - Rules Of Engagement).

Inom vissa områden har SIMSENS samarbetat med andra projekt för att integrera simuleringsmodeller till MSSLab:

- Signalbehandlingsalgoritmer för detektion, målföljning, återigenkänning, sensorstyrning och ruttplanering har i huvudsak tagits fram inom projekten "Signalbehandling för styrbara sensorsystem" och "Autonom spaning". SIMSENS och ARCUS-projektet har i huvudsak stått för integration av dessa algoritmer till MSSLab.
- Vidareutveckling och integrering av laserfederaten till MSSLab har skett i samarbete med projekten "Måligenkänning med högupplösande 3D-avbildande laserradar" och SE-MARK.
- IR-modellering av HKP10 har skett i samarbete mellan projekten SIMSENS, Signaturmodellering och SE-MARK.
- IR-modellering av fordonen BTR70, MTLB, T72 och Volvo V70 har tagits fram i projekten "Svårupptäckt precisionsmålsökare" och "Avancerade målsökare". Vidareutveckling av dessa har skett i SIMSENS.
- En flygmodell för ett UAV-system har levererats av FoT-projektet "Flexibla vapen och skydd".
- Modellen av en militär camp i Mellanösternmodellen har tagits fram i projektet "Nytt Telekonfliktverktyg".

2 Terrängmodeller

För att göra det möjligt att simulera sensorsystem i olika miljöer har arbete utförts i projektet för att förbättra de olika terrängmodellerna. Terrängmodellerna som vidareutvecklats under 2011 är: Norrköping, Kvarn, och Mellanöstern. Arbetet har under 2011 bestått i att materialklassificera de olika modellerna enligt en metodik för materialklassificering som togs fram i SIMSENS-projektet under 2010[1].

2.1 Norrköpingsmodellen

Vid framtagningen av Norrköpingsmodellen har data använts från ett lasersystem och en visuell sensor som båda har varit monterade på ett flygplan. Norrköpingsmodellen innehåller en högupplöst markyta, automatiskt rekonstruerade byggnader, vägnät samt modeller av enskilda träd med korrekta dimensioner inplacerade på korrekta positioner. Den rumsliga upplösningen i modellen är ca 5-10 punkter/m² med en koordinatnoggrannhet på 0.1 m.

Modellen täcker en yta på 3.4×2.9 kilometer av centrala Norrköping och markytan har en texturupplösning på 0.2 meter. Figur 2 visar det ortofoto som använts i modellen och modellens täckningsområde.



Figur 2. Ortofoto som visar Norrköpingsmodellens täckningsområde.

Från kommunens basdata har byggnadernas form och textur bestämts. Väg- och taktexturer har valts automatiskt, baserat på om byggnaden t.ex. är en fabrik eller bostadshus. Höjden på byggnaderna har extraherats från laserdata. Vägnätet kommer också från kommunens basdata. I figur 3 visas en del av Norrköpingsmodellen och en detaljerad bild visas i figur 4.

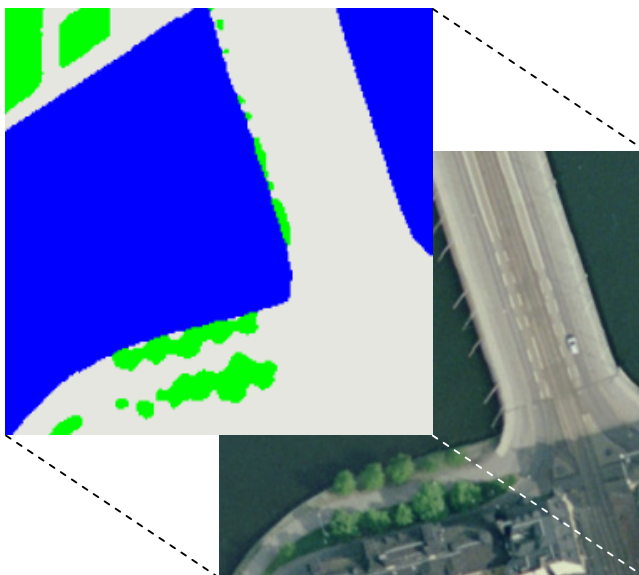


Figur 3. Del av Norrköpingsmodellen.



Figur 4. Detaljbild av Norrköpingsmodellen.

Det finns även en materialklassad modell som används vid sensorsimulering av IR- och radarsensorer. Ett litet utsnitt av den materialklassade Norrköpingsmodellen visas i figur 5. I figuren är färgen satt för de olika materialklasserna enligt följande: blå avser vatten, grön avser vegetation och grått avser övrigt. Ingen skillnad har gjorts på till exempel asfalt och hustak eftersom fristående husmodeller används ovanpå den klassificerade marktexturen.



Figur 5. Uppdelning av marktextur i lager baserat på resultat av klassificering av markpixlar. Blå färg avser vatten, grönt avser vegetation, grå färg avser övrigt.

Figur 6 visar hur en simulerad LWIR-sensor registrerar ett område i Norrköping, simulerad med SE-WORKBENCH-EO. Upplösningen på den simulerade sensors är 1024×1024 pixlar.

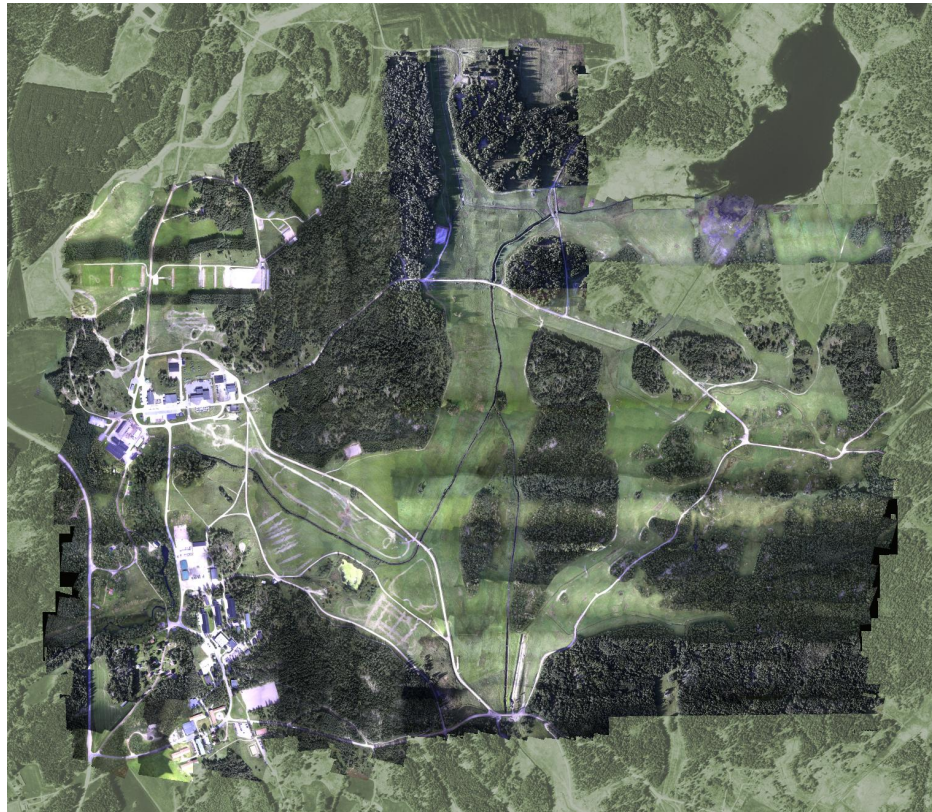


Figur 6. Registrering från en simulerad LWIR-sensor över ett område i Norrköping. Sensorupplösningen är 1024×1024 pixlar.

2.2 Kvarnmodellen

Kvarnmodellen representerar ett modellerat avsnitt 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 omgivningsmodellen skapats. Kvarnmodel-

len innehåller en högupplöst markyta, automatiskt rekonstruerade byggnader och modeller av enskilda träd med korrekta dimensioner inplacerade på korrekta positioner. Den rumsliga upplösningen i modellen är ca 5-10 punkter/m² och med en koordinatnoggrannhet på 0.1 m. Detta kan jämföras med Lantmäteriverkets standardprodukt med 50 meter mellan höjdvärdena. Modellen täcker en yta på 2.8×1.6 kilometer. Figur 7 visar det ortofoto som använts i modellen och modellens täckningsområde.



Figur 7. Ortofoto som visar Kvarnmodellens täckningsområde.

Högupplöst visuella data och IR-data finns även insamlat över det laseruppmätta området. Dessa data har 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 8 och figur 9.

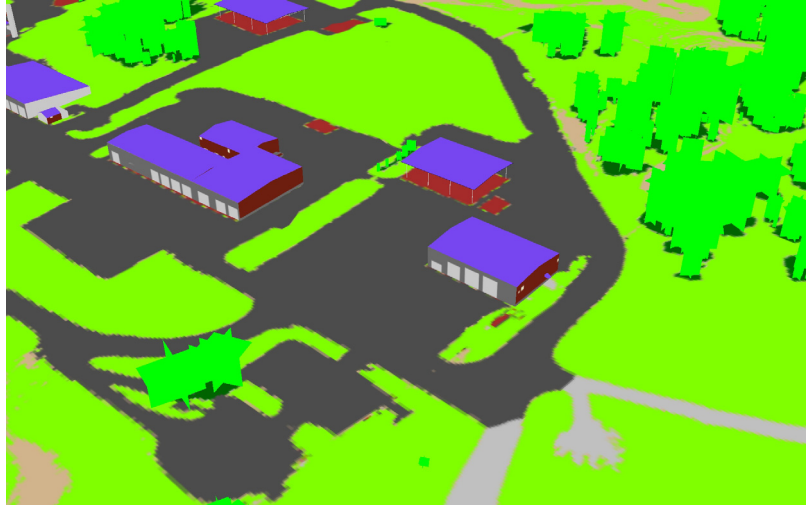


Figur 8. Bilden visar en del av Kvarnmodellen med uppmätt visuell textur.



Figur 9. Bilden visar en del av Kvarnmodellen med uppmätt IR-textur.

Det finns även en materialklassad modell som används vid högkvalitativa sensor-simuleringar av IR- och radarsystem, vilket visas i figur 10. Marken är klassificerad med följande material: gräs, asfalt, undervegetation, lera, berg och vatten. Byggnaderna är klassificerade med materialen glas, betong, trä, metall samt tegel. Klassificeringen har sedan kombinerats med det högupplösta visuella data för att bestämma finare materialupplösning samt blandningar av material. Figur 11 visar en registrering från en LWIR-sensor simulerad i SE-WORKBENCH-EO.



Figur 10. Bilden visar en del av Kvarnmodellen med materialklassad textur. Färgerna i bilden motsvarar olika materialtyper.



Figur 11. Bilden visar en registrering från en LWIR-sensor över en del av Kvarn. SE-WORKBENCH-EO har använts vid simuleringen.

De tre omgivningsmodellerna är alla geografiskt och geometriskt korrelerade.

För varje träd i det uppmätta området har information om position, höjd, vidd och trädslag extraherats. Modeller av enskilda träd har sedan placerats in i omvärldsmodellerna med korrekta dimensioner och på korrekta positioner. Ett problem med omvärldsmodeller med noggrann beskrivning av vegetation är att dessa blir mycket stora och kräver mycket minne. Förenklade trädmodeller har använts i modellen för att undvika tunga beräkningar.

2.3 Mellanösternmodellen

Mellanösternmodellen som visas i figur 12 är en geotypisk terrängmodell, till skillnad från Kvarn- och Norrköpingsmodellen som är geospecifika modeller. Geospecifika modeller är modellerade för att efterlikna verkligheten, vilket gör det möjligt att validera genomförda simuleringar mot riktiga mätningar. Geotypiska modeller försöker efterlikna en typisk miljö, vilket är bra då man vill studera generella frågeställningar.

Mellanösternmodellen består till största delen av öken, men den innehåller även en stadsmiljö samt en militär camp. Den uppbyggda stadsmiljön är baserad på en inköpt modell av en Mellanöstern-liknande stad. Modellen täcker en yta på 5×5 kilometer.



Figur 12. Mellanösternmodellen innehåller en typisk stadsmiljö för ett område i Mellanöstern.

Den militära campen har tagits fram i projektet "Nytt Telekonfliktverktyg" [2] och kallas Camp GENESIS. Inspirationskälla för denna Camp har varit den svenska Campen i Mazar-e Sharif. Modellen är fiktiv, men byggnader och annan geometri är inspirerad av bilder från den verkliga platsen, se figur 13.

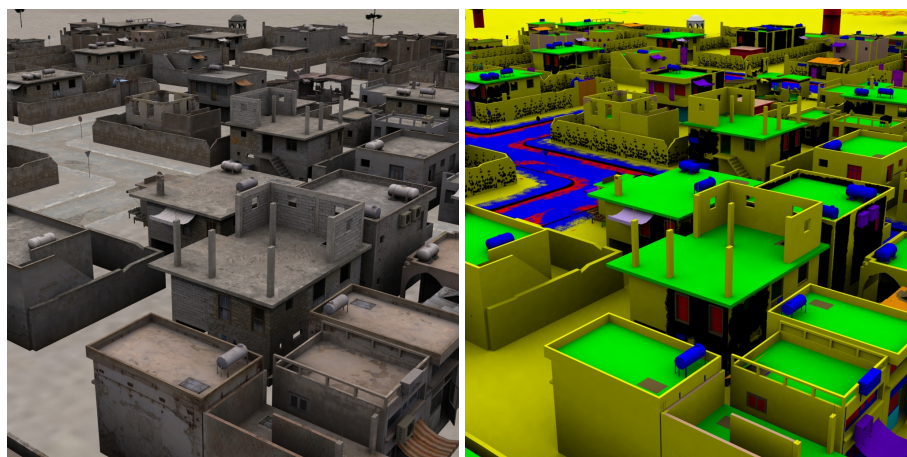


Figur 13. Camp GENESIS.

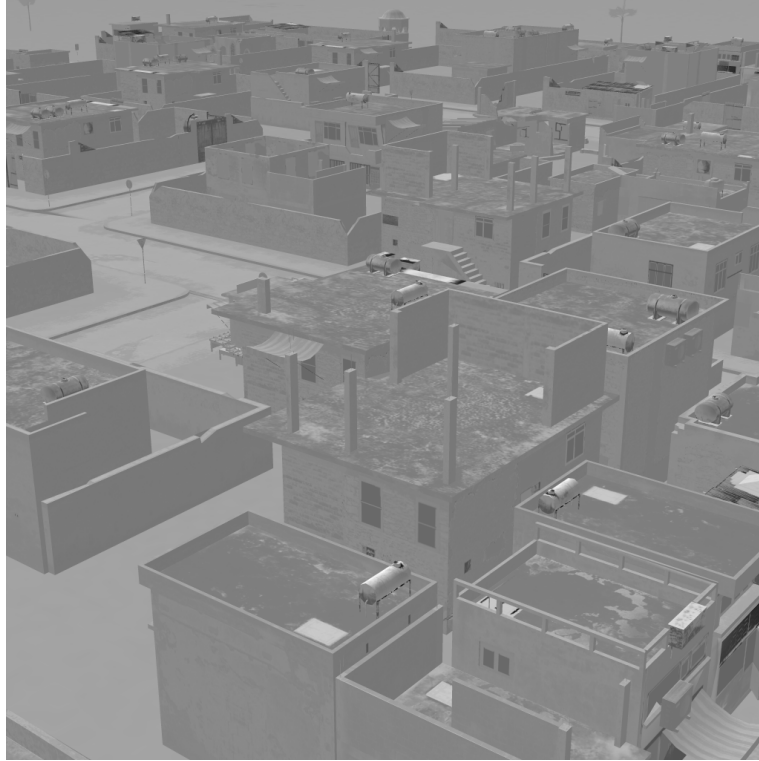
I Mellanösternmodellen har vägar mellan olika stadsmoduler byggts in och längs med dessa vägar har även mindre byggnader placerats för att på så sätt kunna utnyttja en större del av omvärlden utan att det skall upplevas som en kuliss. Vegetation i form av olika träd och palmer finns utplacerade i terrängen. Dessa består av väldigt enkla geometrier för att inte belasta visualiseringar i realtid och är till för att ge mer realism i omvärldsmodellen. Träden är skapade i verktyget RealNat Premium. Positionerna för träden har slumpats ut över området och sedan har de träd som hamnat på icke önskade positioner tagits bort. Slumppositionerna har genererats med ett plugin som gör det möjligt att måla ut områden där instanser av objekt (i det här fallet träden) skall hamna.

Både höjd- och rasterdata är framtaget på syntetisk väg. Storleken på omvärldsmodellen är 5×5 km och höjddata har en upplösning på 0.1 punkter/m. Rasterdata har en relativt låg upplösning på 0.2 pixlar/m och för att väga upp detta har olika detaljtexturer applicerats med avseende på terrängtyp.

Även för Mellanösternmodellen har materialklassade texturer tagits fram. Detta har skett genom att manuellt materialsätta samtliga texturer som används i modellen. Texturerna har modifierats till färgmaskade områden för varje enskild bild, där en viss färg motsvarar en materialklass. Figur 14 visar ett exempel på hur färgmaskningen kan se ut över ett område i modellen. Figur 15 visar en simulering av en LWIR-sensor med SE-WORKBENCH-EO med den materialklassade modellen.



Figur 14. Exempel på färgmaskning över ett område i Mellanösternmodellen.



Figur 15. Ett område i Mellanösternmodellen simulerad med SE-WORKBENCH-EO.

3 Objektmodeller och animering av människor

Vid värdering av sensorsystem med simuleringsbaserade metoder är det mycket viktigt att modellerna för såväl bakgrund som objekt är av hög kvalitet. Omfattande arbete har utförts för att vidareutveckla ett 3D-modellbibliotek med olika objekt och människor, som sedan kan används för att beräkna realistiska signaturer i olika scenarier.

3.1 Objektmodeller med visuell textur

Några av de tillgängliga modellerna med visuell textur visas i figur 16, [3]. Arbetet med dessa modeller har i huvudsak bestått i att manuellt skala om dessa modeller och se till så att deras koordinatsystem är korrekt orienterat.



Figur 16. Exempel på några tillgängliga objektmodeller med visuell textur.

3.2 Beräkning av IR-signaturer av människor

En förenklad modellering av signaturen från fem olika människor har genomförts (se figur 17). Människomodellerna är uppdelade i tio olika delar. Materialegenskaperna för klädesplaggen har i denna simulering antagits då vi för tillfället inte har tillgång till en materialdatabas för olika kläder.



Figur 17. Fem modeller av personer som har använts vid beräkning av IR-signaturer.

I figur 18 visas beräkningar av IR-signaturen av människor med en förenklad ansats där kroppstemperaturen är låst till 37° C med en hög värmeövergångskoefficient. Modellen är solbelyst från ena sidan och den har olika tjocka luftlager mellan kropp och kläder.

En skillnad vid simulering av signaturer från människor är att den termiska historiken inte är lika viktig som vid simulering av signaturer på stridsfordon eller fartyg. Ett kortare tidsintervall är dock lämpligt för att eventuell påverkan av soluppvärmning ska bli korrekt. Ett tidsintervall på tio minuter är troligen rimligt där personen kan ges en tidsupplöst beskrivning i olika gångriktningar.



Figur 18. Modellering av människor med en förenklad ansats där innertemperaturen är låst till 37° C.

Resultaten från människomodelleringen kan i framtiden förbättras genom att använda RadTherm-modulen "human comfort module". Denna innehåller bland annat en materialdatabas för olika kläder samt en fysikalisk modellering av blodomlopp och svettning m.m. Man kan då modellera människorna inuti t.ex. ett fordon eller inne i ett hus för att se hur personen upplever komforten med avseende på temperaturen. T.ex. kan effekten av aircondition eller olika färger eller ytbeläggning på glas modelleras i kombination med att en människa finns i fordonet eller byggnaden.

3.3 Animering av människor

Simulering av sensordata med animerade människor är en verksamhet som påbörjades 2008. För att göra detta möjligt har HLAS, High Level Animation System, tagits fram inom SIMSENS-projektet. Med HLAS underlättas integrationen av animerade sekvenser för olika typer av simuleringsramverk. För att skapa och visualisera skelettbase-erade animationer finns det flera verktyg, men utbudet är inte lika stort då man vill kunna styra dessa animeringssekvenser på en högre abstraktionsnivå. Ett exempel på ett sådant fall kan vara att en användare vill förflytta en karaktär från en punkt till en annan utan att behöva ta hänsyn till hur terrängen ser ut. För att en sådan förflyttning ska utföras på ett trovärdigt sätt behövs ett stöd för vilka animationer som ska spelas

Ett större bibliotek av människor har tagits fram utifrån grunddata som är köpt från Rocketbox Libraries. Det finns 21 manliga och 21 kvinnliga modeller med 2 olika form av klädnad för varje. Utöver dessa finns även två olika modeller med arabiskt utseende och med olika texturer finns dessa som 6 varianter. Alla manliga (inklusive de arabiska) respektive alla kvinnliga karaktärsmodeller använder sig av samma animations- och konfigurationsfiler och det går således att byta ut enskilda karaktärer utan att ändra hela scenarion. Av totalt 500 olika animationer som följde med originaldata har ca 140 st exporterats till HLAS.

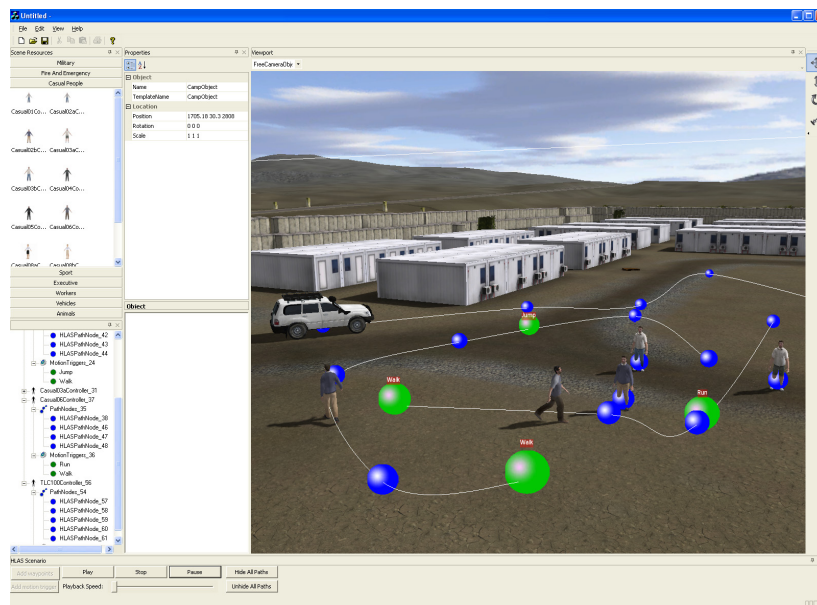
Ett bibliotek med 50 olika soldater från Rocketbox Libraries har köpts in. I ett framtida arbete kommer även dessa modeller att exporteras till HLAS.



Figur 19. Modeller som idag används vid animering av människor.

Grundtanken med designen av HLAS är att skapa ett plattformsberoende animations system som på ett enkelt sätt ska kunna utvidgas, integreras och anpassas för olika typer av simuleringar.

Grundscenariomodellen i HLAS är baserad på brytpunktsnavigering, vilket visas i figur 20. Denna navigering genereras genom att användaren först anger ett antal brytpunkter därefter några rörelsekommandon, s.k. *motion triggers*. En *motion trigger* är aktiv och förblir aktiv så länge ingen ny *motion trigger* har passerats eller att dess livslängd förlöpt.



Figur 20. Scenario med brytpunktsnavigering.

För att enkelt kunna skapa scenariofiler har ett scenarioverktyg utvecklats till HLAS. I scenarioverktyget kan användaren placera brytpunkter och *motion triggers* direkt i den omgivning där scenariot skall utspelas. Användaren har även möjlighet att spela upp, pausa och spola fram scenariot samtidigt som scenariot redigeras.

Då man vill simulera en större grupp av människor blir metoden ovan olämplig då varje karaktär behöver regisseras individuellt. För detta ändamål har mer autonoma beteenden tagits fram. Istället för att jobba med individuella karaktärer används en gruppkontroll. Gruppkontrollen kan ses som en ledare för gruppen som bestämmer vilka individer som ska ingå i gruppen, vilka resurser som är kopplade till gruppen samt vilket beteende gruppen ska använda. Exempel på beteenden kan vara att följa en brytpunktsbana, röra sig slumpmässigt inom ett område eller att förfölja en annan karaktär. Karaktärernas förflyttning simuleras med hjälp av styrkrafter, s.k. *forceactions*. Dessa krafter på appliceras på karaktären utifrån prioritet och vikt. Ett exempel kan vara att en karaktär i första hand vill undvika att kollidera med terrängobjekt, i andra hand undvika att kollidera med dynamiska objekt (exempelvis bilar eller andra karaktärer) och i tredje hand röra sig mot en brytpunkt.



Figur 21. Styrbeteenden för kollisionshantering visualiserat med röd cirkel, vägsökning visualiserade som röda linjer och resulterad kraft i blått.

Beteenden i HLAS skapas och konfigureras med hjälp av lua-skript som sedan kopplas till en gruppkontroll i scenarioeditorn. Eftersom lua är ett skriptspråk kan beteenden ändras under scenarioskapandet utan att någon programkod behöver kompileras om, vilket är ett krav då beteenden varierar kraftigt mellan olika scenarier. När man skapar ett lua-skript-beteende är man dock begränsad till några grundläggande *force actions*, dessa är i dagsläget listade nedan:

- *MapSteerAction* - Kollisioner hantering med en binär karta
- *NavMeshAction* - Kollision hantering med en navigations mesh
- *SteerToAvoidNeighborsAction* - Undviker kollision med andra karaktärer
- *SteerToAvoidObstaclesAction* - Undviker kollision med hinder
- *WanderAction* - Vandrar planlös
- *PathFollowAction* - Följer en brytpunktsbana
- *FlockAction* - Skapar ett typiskt flockbeteende
- *PursuitAction* - Jagar en annan karaktär
- *SpeedAction* - Försöker hålla en given hastighet
- *DefinedFormationAction* - Gruppen försöker hålla en given formation
- *PedestrianAction* - Går till slumpmässig position inom en navigation mesh och använder navigations-meshen för att hitta kortastevägen
- *GoToPositionAction* - använder navigationsmeshen för att först hitta en väg till en given position och sedan följer denna väg/brytpunktsbana

- *SteerToAvoidExternalAction* - Undviker kollision med externa entiteter såsom andra karaktärer eller fordon.

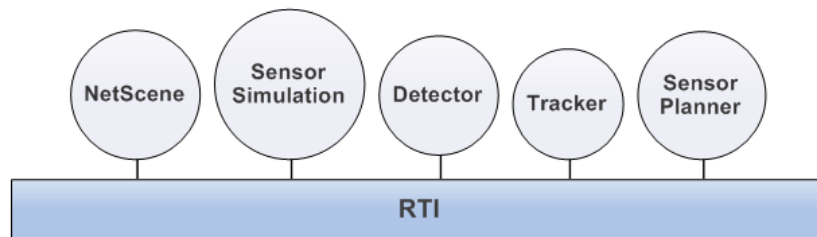
Dessa *force actions* är implementerade i c++ av prestanda skäl och kombineras sedan i ett lua-skript för att skapa ett dynamiskt beteende.

Ett exempel kan vara att en grupp till en början följer en brytpunktsbana (*PathFollowAction*), men då någon i gruppen är i närheten av en given entitet växlas beteendet till att jaga denna person (*PursuitAction*).

4 Distribuerade simuleringar

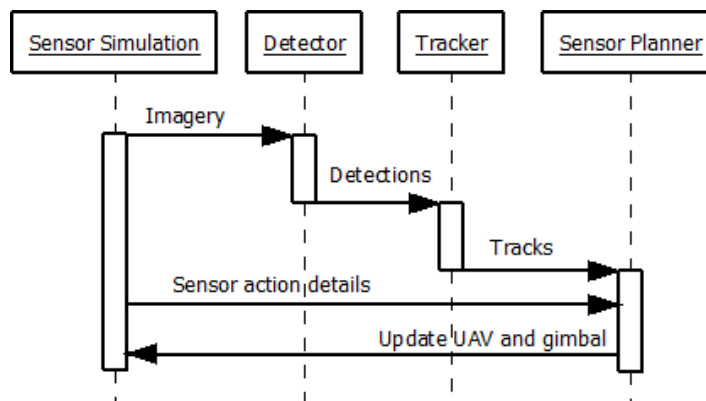
Simuleringsmiljön MSSLab är designad för att vara en distribuerad simuleringsmiljö, vilket gör den modulär och flexibel. Den distribuerade simuleringen i MSSLab bygger på MOSART[18], som är ett ramverk för High Level Architecture (HLA) [19]. HLA-simuleringar kallas federationer och federater är de distribuerade delarna i simuleringen.

MSSLab bygger upp federationer på ett sätt som efterliknar ett sensorsystem. Fördelen med detta är att man kan ersätta en sensoralgoritm mot en annan bara genom att ansluta en annan version av t.ex. detektor-federaten (se figur 22).



Figur 22. Exemple på en federation i MSSLab.

Delarna i simuleringen utbyter data genom en Federation Object Model (FOM). MSSLab har utökat existerande standard med information nödvändig för att de olika delarna av sensorsystemet ska kunna kopplas ihop i en sensoralgoritmkedja (se figur 23).

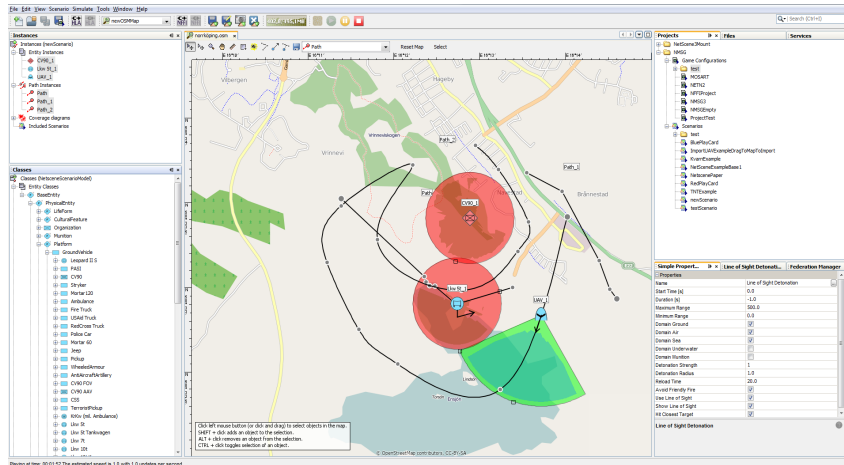


Figur 23. Informationsutbyte i en typisk MSSLab-federation.

Sekvensdiagramet i figur 23 beskriver hur sensorinformation kommuniceras och förädlas. En federat, *Sensor Simulation*, genererar sensordata, i det här fallet en visuell bild, som skickas till prenumeranter i federationen. Federaten *Detector* genomför detektion i bilden och genererar sedan detektioner. Detektionerna filtreras sedan till målspar i *Tracker* federaten. Målspar och information om sensoraktioner kan sedan användas av en *Sensor Planner* federat för att styra *Sensor Simulation* federaten.

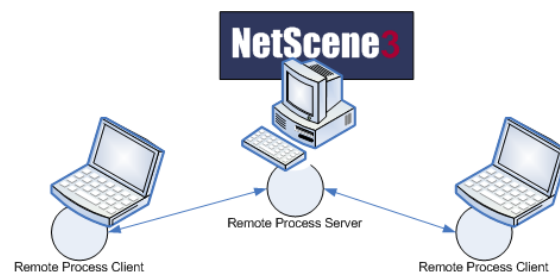
4.1 Scenarioverktyget NetScene

NetScene[20] är en scenarioeditor och en scenariomotor för distribuerade simuleringar. I detta verktyg kan olika scenarier specificeras och objekten i simuleringen parametersättas. Scenarioinformationen distribueras sedan till övriga simuleringskomponenter (federater) som sedan sköter själva simuleringarna av resp. objekt/företeelse. Detta medger en mycket stor flexibilitet där en viss simuleringskomponent kan bytas ut mot en mer lämplig sådan utan att scenarioverktyget behöver förändras. NetScene innehåller rörelsemodeller, enkla sensormodeller, bekämpningsmodeller etc., vilket visas i figur 24.



Figur 24. Beteendemodeller visualiserade i NetScene, bekämpningsmodell i rött, sensormodell i grönt, rörelsemodeller enligt utritade trajektorier.

NetScene skapar simuleringen på två sätt; antingen skapar NetScene federationen genom att starta federater med Remote Process Manager, eller så skapar NetScene scenariot genom att skicka så kallade RemoteCreate-interaktioner till deltagande federater.



Figur 25. Klienter kopplar upp mot NetScene genom Remote Process Server.

NetScene innehåller en Remote Process Server till vilken flera klienter kan koppla upp sig. NetScene kan sedan starta upp jobb på uppkopplade klienter.

4.2 MSSLab-federater

I MSSLab finns det idag många olika federater. Några federater som används ofta är:

- *DetectionFederate* är en MATLAB-federat som analyserar sensordata och producerar detektioner till federationen.

- *EntityController* är en MATLAB-federat som styr enheter i simuleringen enligt förutbestämda banor. NetScene skapar entiteter i EntityController med hjälp av RemoteCreate-interaktioner.
- *HiFiEngineFederate* är en federat för realistisk plattformssimulering på marken och i luften. Federaten kan simulera markfordon såsom stridsvagnar, pansarterrängbilar, anläggningsmaskiner och personbilar. I luften kan helikoptrar och flygplan simuleras.
- *HLASFederate* är ett tillägg till HLAS scenario verktyg som möjliggör att simulering av mänskligt beteende kan distribueras över HLA.
- *SceneServerFederate* är en Java-federat som skapar en koppling till SceneServer. SceneServerFederate reflekterar information, t ex en pansarvagn, från den distribuerade simuleringen till SceneServer. SceneServerFederate kan även initiera HLA:s sekvenser i SceneServer. SceneServerFederate kan sedan publicera sensordata i form av bilder till federationen.

4.3 Batchkörningar med MSSLab

För att kunna få fram tillförlitliga sensorprestanda krävs ofta ett statistiskt underlag, vilket kräver många simuleringar. Detta har åstadkommits genom att utnyttja NetScenes förmåga att göra batchkörningar. Vid batchkörningar skapas en federation med olika simuleringskomponenter (federater). Federationen körs under en förutbestämd tid och sedan startas federationen om igen. Olika parametrar kan varieras under batchkörningarna. I SIMSENS-projektet har EntitySequenceNumber varierats mellan körningar. EntitySequenceNumber används av EntityController för att välja olika flygbanor att styra entiteterna efter.

5 Simulering av sensorsystem

I SIMSENS-projektet arbetar vi med att simulera olika sensorsystem i olika miljöer, vädersituationer och tidpunkter på året. I detta avsnitt kommer vi att beskriva arbetet som har bedrivits under 2011 med de olika sensorsimuleringsmodellerna.

5.1 Visuella simuleringar av ett marint scenario

Visuella simuleringar av ett marint scenario har utförts med SceneServer. SceneServer [4] är ett program för realtidsvisualisering av syntetiska omgivningar och är baserat på OpenSceneGraph [5]. SceneServer simulerar modeller baserade på 3D-geometri och texturer.

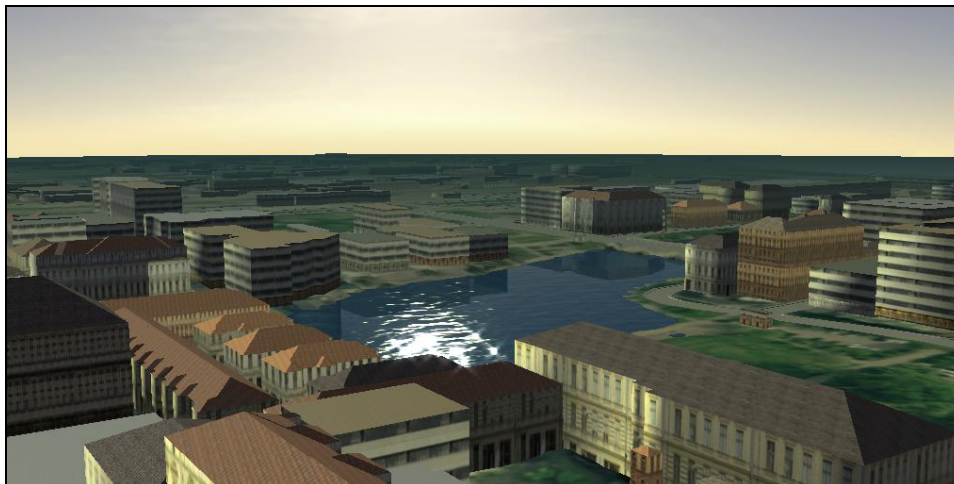
För vattensimuleringar används modulen osgOcean [6]. Modulen möjliggör många simuleringsmöjligheter med parametrar så som vindhastighet, skumbildning, refraktion (är ljusstrålars brytning vid övergången mellan två ämnen med olika brytningsindex), spegelbilder, vattendjup och solreflektioner. I figur 26 till figur 28 visas några simuleringar som har utförts. Norrköpingsmodellen har använts som terrängmodell.



Figur 26. Visuell simulering av lugnt vatten med reflektioner från Norrköpingsmodellen.



Figur 27. Visuell simulering av öppet hav med vågor.



Figur 28. Visuellt simulering av Norrköpingsmodellen med solreflektioner.

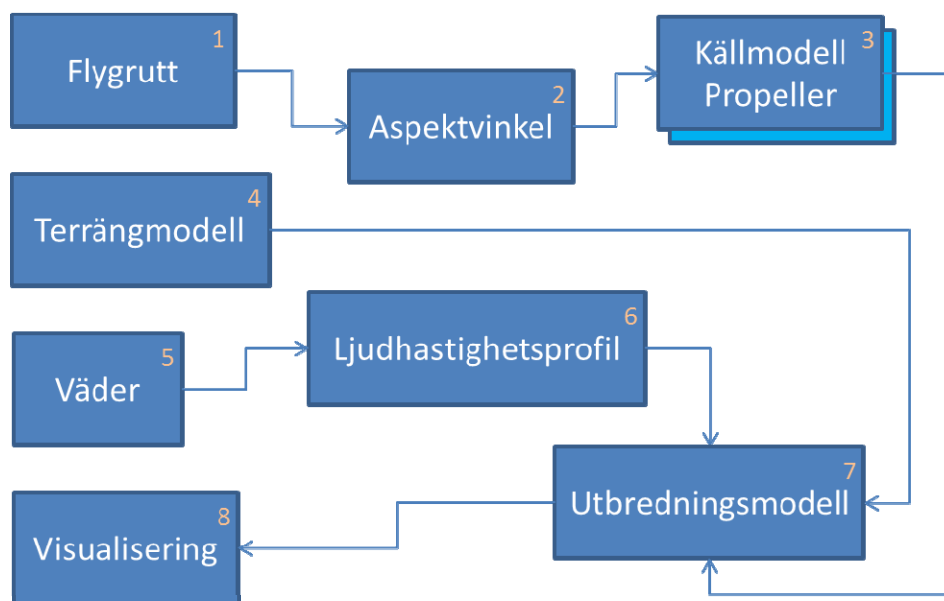
5.2 Simulering av ljudutbredning från UAV-system

Arbetet har bestått i att vidareutveckla FOI-AkustikSim för att kunna visa var på marken man kan registrera ett UAV-system akustiskt med hjälp av hörsel eller sensorer. Simulatorens tar hänsyn till aktuellt väder med parametrar som påverkar den akustiska utbredningen, exempelvis: vind, temperatur, nederbörd och källans spektrala egenskaper och dess direktivitet.

Akustiska modeller finns framtagna för olika luftfarkoster som framdrivs med propeller (flygplan och UAV) och i en fortsatt kan även modeller utvecklas även för jet-turbin och helikopterrotor. Även andra ljud som skott från vapen, detonation från sprängämnen, motorbuller från fordon och däckljud etc. kan vara aktuella att modelleras för att användas som källa vid beräkning av ljudutbredning.

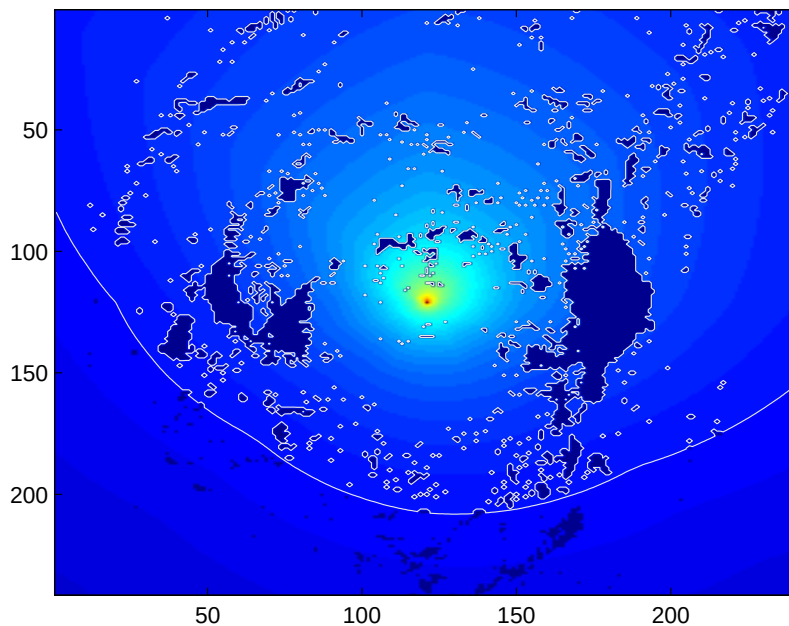
I blockschemat i figur 29 visas arkitekturen för en tidig version av ljudutbredningssimulatorens som baserats på en motsvarande simulator för markfordon. Denna var framtagna i ett projekt för demonstration och visualisering av hotområden för stridsfordon.

1. Flygrutten beskrivs i formaten: RT90, SWEREF99, WGS84 eller liknande samt höjd över markytan i meter.
2. Aspektvinkeln till ljudkällan relaterar till farkostens längdaxel och attityd vilket ger ljudkällans karaktäristik rätt orientering.
3. I första fasen behandlas flygande farkoster med propellerframdrivning.
4. Terrängmodellen är baserad på höjd i meter över havsytan och kan beskrivas med formaten i punkt 1.
5. De väderbaserade parametrar som påverkar ljudutbredningen beräknas beroende på observationer, mätvärden och modeller.
6. Ljudhastighetsprofilen beräknas på tillgängliga data och beskrivs som hastighetskomponenter i nordlig (0 grader) och östlig (90 grader) eller latitud och longitud med höjdangivelse i meter.
7. Utbredningsmodellen sammanväger väderberoende parametrar (t.ex. refraction och "scattering"), källa och terrängmodell.
8. Visualisering sker genom utskrift i Matlab eller export till annan programvara.



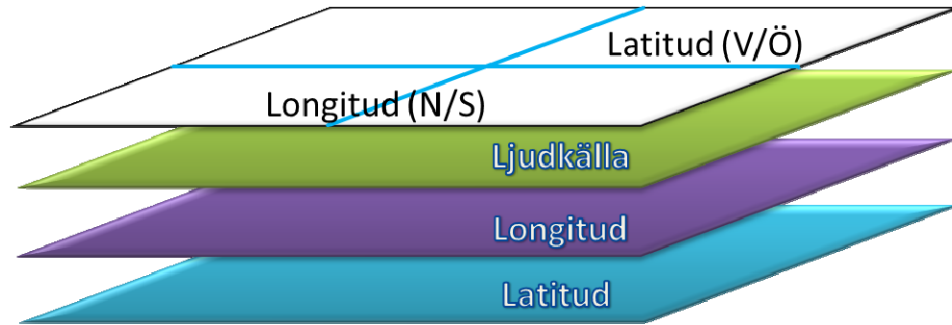
Figur 29. Blockschema för simuleringsmodellen.

Figur 30 visar ett exempel från simulering av akustisk utbredning från fordon för demonstration och visualisering av hotområden. Upptäcktsavståndet har markerats med den vita linjen som representerar ett gränsvärde för upptäckt med hörsel. De ursprungliga algoritmerna baseras på "ray tracing" från källa till sensor och blir relativt beräkningsintensiv. Eftersom relevanta väderdata är svårt att erhålla bidrar detta till en tidskrävande simulering med stor osäkerhet i resulterande ljudfält.



Figur 30. Exempel från simulering av akustisk utbredning från fordon för demonstration och visualisering av hotområden. Grid: 50 m vilket motsvarar en radie på ca: 6 km.

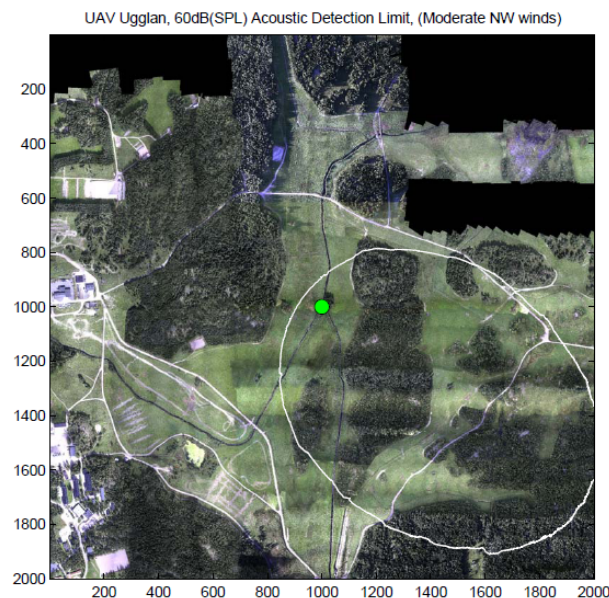
Då simulatorn som utvecklades för markfordon har visat sig mindre användbar för simulering av luftburna farkoster har denna utbytt mot en simulator som bygger på empiriska modeller där ljudutbredning som kopplas till olika väderlekar, vindstyrka och riktning. Den nya simulatormodellen innefattar sammanvägning av väder (5), ljudhastighetsprofil (6) och utbredningsmodell (7) där ljudfältet kommer att vägas gentemot uppmätta parametrar och interpoleras mellan dessa.



Figur 31. Matriser för beräkning av resulterande ljudfält beroende på orientering av bland annat ljudkälla, vindstyrka och vindriktning.

Vid simulering av ljudfält i den nya modellen kommer inte lokala avvikelser orsakade av akustiska hinder motsvarande byggnader och terräng utföras då den funktionen kräver omfattande beräkningskapacitet vid hög upplösning i ett stort område.

Ett exempel från en simulering av upptäcktsavståndet för UAV-systemet Ugglan i den nya simulatorn visas i figur 32. Vinden påverkar ljudutbredningen och ljudet dämpas mer mot än med vindriktningen och därmed blir upptäcktsavståndet till källan kortare respektive längre.

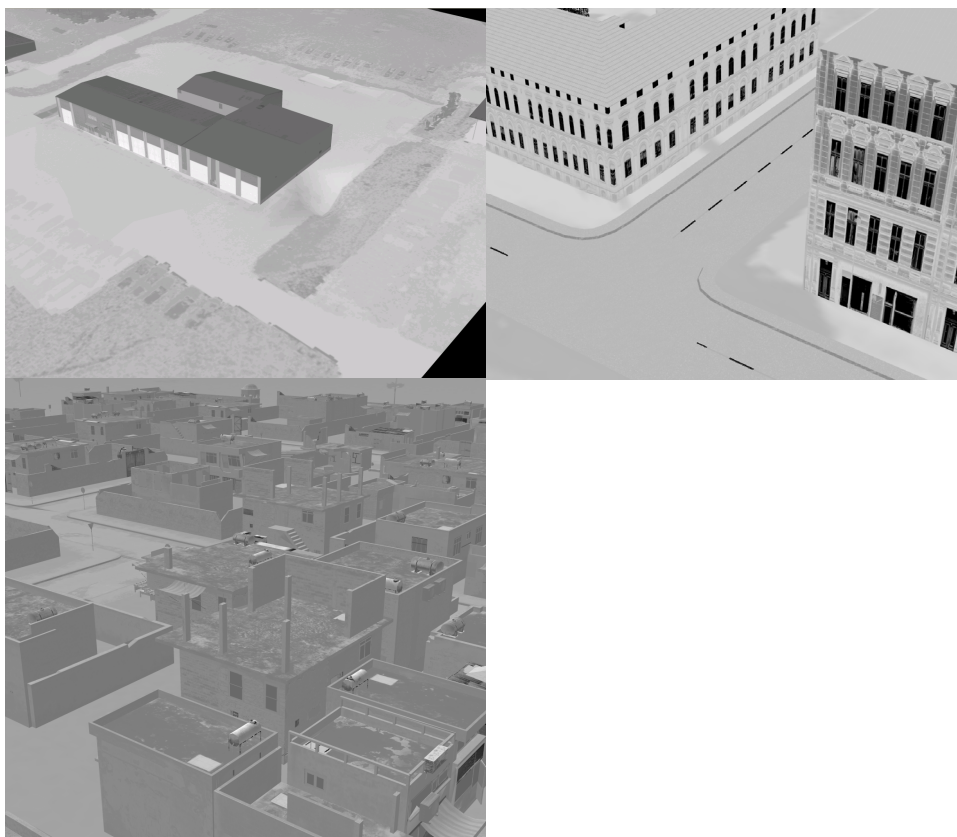


Figur 32. Upptäcktsavstånd (vita linjen) för UAV-systemet Ugglan (grön punkt) på höjden 900m vid nordvästliga vindförhållanden. Den översta bilden visar ett visuellt ortofoto och den nedersta bilden visar en höjdmödel för terrängen. Axlarnas enheter är meter.

5.3 Simulering av IR i olika miljöer

För noggranna simuleringar av optiska sensorer i stora scener används SE-WORKBENCH-EO (Synthetic Environment Workbench). SE-WORKBENCH-EO är framtaget av Oktal SE i Frankrike och har använts på FOI sedan 2009. I SE-WORKBENCH-EO ingår applikationen SE-RAY-IR som kan simulera sensorer inom våglängdsområdena: UV, Visuellt, NIR och TIR. Scenarierna kan vara både statiska och dynamiska. I SE-RAY-IR finns fysikaliskt baserade modeller för väder, termodynamik för beräkning av fysikalisk ytemperatur, avancerade renderingsalgoritmer för modellering av spridning, transmission, reflektans och ljusabsorption.

Arbetet har under 2011 bestått i att materialklassificera terrängmodellerna som används i MSSLab, dvs. Kvarnmodellen, Norrköpingsmodellen och Mellanösternmodellen, vilket är beskrivet i kapitel 2. IR-simuleringar med SE-WORKBENCH-EO i omvärldsmodellerna Kvarn, Norrköping och Mellanöstern visas i figur 33.



Figur 33. IR-simuleringar med SE-WORKBENCH i omvärldsmodellerna Kvarn, Norrköping och Mellanöstern.

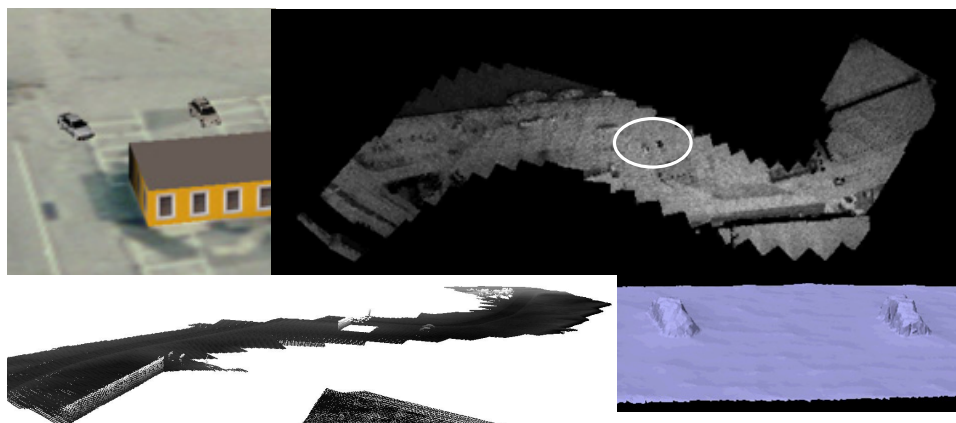
En begränsning med SE-RAY-IR är att termodynamiska beräkningar endast utförs i en dimension. Detta medför att värmeflödet till och från en yta endast beräknas i djupled. För att få en mer detaljerad simulering av värmeledning, strömning och strålning från olika objekt i en scen kan resultat hämtas från RadThermIR (avsnitt 3.2) som kan utföra termodynamiska beräkningar i tre dimensioner.

5.4 Simulering av 3D-avbildande laserradar

Lasersimulatorens *FOI-Ladarsim* som har använts i projektet har vidareutvecklats av bl.a. FoT-projekten ”Måligenkänning med högupplösande 3D-avbildande laserradar” och SIMSENS. Senaste året har prestandan och flexibiliteten med lasermodellen förbättrats, främst genom en efterbehandling av data som ger globalt refererade mätresultat, exempelvis i form av 3D-koordinater inklusive intensitetsvärden. Detta gör att resultat från simuleringar av dynamiska förlopp är möjliga att fusionera, både om sensorn rört sig och/eller om målet har rört sig.

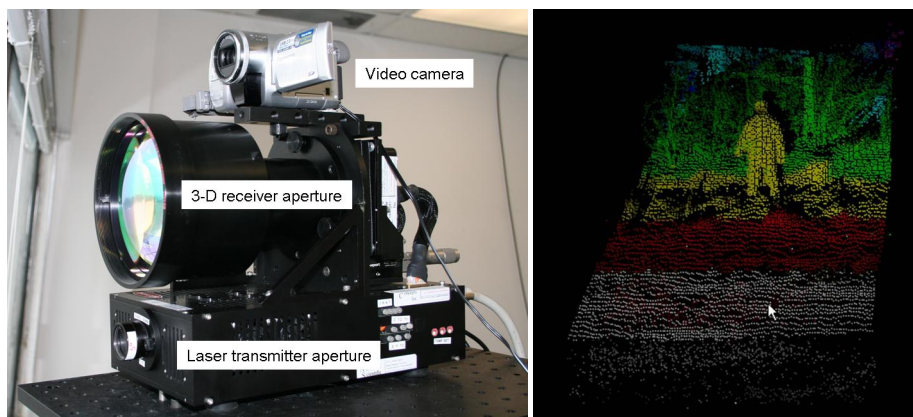
Den stora nyttan med detta är att simuleringar kan utföras i samma scenario som de övriga integrerade sensormodellerna och att alla nödvändiga operationer kan styras på ett enhetligt sätt.

Figur 34 visar resultatet av en lasersimulering i ett scenario med två stycken bilar, en Toyota RAW4 och en BMW E36, parkerade utanför en byggnad i centrala Norrköping. I detta fall har simuleringsparametrar valts för att efterlikna den amerikanska sensorn ASC Flash 3D [9][7] som kan registrera en avståndsbild med 20 stycken avståndsluckor per utsänd laserpuls. Att registrera hela bilden med samma puls är möjligt eftersom laserdetektorn i ASC Flash 3D består av en array av 128×128 detektorer som var och en direkt mäter upp den mottagna vågformen. Data från sensorn är sedan processad till att ge avståndsdata i varje bildelement, och även positionera dessa globalt. I nedre högra delen av figur 34 visas resultatdata triangulerad exempelvis inför en matchning mot en fordonsdatabas.



Figur 34. Simuleringsresultat från ett scenario med två stycken bilar parkerade bredvid ett hus i Norrköping. Överst till vänster visas en bild av scenariot som renderats med SceneServer. De övriga bilderna visar simuleringsresultatet från laserradarn där bilden överst till höger visar punktmolnsdata sett uppifrån och nedre vänstra bilden visar data sett från öster. Bilderna visar en sammanläggning av 26 laserradar-”bilder” insamlade vid en simulerad förbiflygning. Nedre högra bilden visar triangulerade data, där de två fordonen exempelvis kan matchas mot en databas.

Figur 35 visar ett foto på 3D-laserradarn ”ASC Flash 3D” och 3D-mätdata från denna sensor föreställande en människa i skogsmiljö. I tabell 1 redovisas sensorspecifika parametrar.



Figur 35. Vänster: Lasersensorn ASC Flash 3D. Höger: Avståndsbild som visar en människa i skogsmiljö upptagen med ASC Flash 3D.

Tabell 1. Laserparametrar

Våglängd	1.57 μm
Maximal pulsrepetitionsfrekvens (frame rate)	30 Hz
Sensorstorlek	128×128 pixlar
Pixelstorlek (pitch)	100 μm
Avstånd	3 – 1500 m
Avståndsnoggrannhet	± 7.6 cm
Synfält	0.5 – 9 °
Dimensioner (bredd x höjd x djup)	15 × 15 × 28 cm
Vikt	< 5.7 kg
Elektrisk effekt	< 400 W
Beam Profile	Rectangular

3D-lasermodellen som har integrerats i MSSLab har utvecklats vid FOI under flera år. Programmet är i dagsläget baserat på SceneServer, vilket ger full insikt och kontroll av koden och möjligheter att skraddarsy applikationer och anpassningar. Programmet bygger på en ray-tracing-liknande princip och är uppdelat i fyra huvuddelar; laserkällan, scenen, atmosfären samt mottagaren [8,9,10].

För laserkällan finns möjlighet att använda en godtycklig analytisk eller diskret beskrivning av pulsformen, såväl i temporal (i ljusets utbredningsriktning) som spatiell (tvärs emot utbredningsriktningen) utbredning. I denna fas bestäms även vilken underliggande upplösning som ska användas för att få med egenskaper mindre än ett bild-element.

För att generera ett tidsupplöst eko sänds simulerad laserenergi mot scenen i form av en stor mängd strålar. Det modellerade ekot genereras sedan genom att analysera hur varje stråle påverkas på sin färd till och från scenen. För varje stråle bestäms avståndet till respektive del av scenen samt reflekterad ljusenergi. Den reflekterade ljusenergin

bestäms genom att analysera vilket material och i vilken vinkel som strålen träffar respektive material.

När optiska vågor färdas genom atmosfären påverkas de av turbulens och partiklar i luften. Atmosfärseffekterna har delats upp och simuleras var för sig. Bland annat sker en dämpning, en scintillation och en utsmetning av energin samt även en strålvandring som får ljusets centroid att avvika från sin utbredningsriktning. Modeller för att simulera vådrets påverkan på lasersystemet saknas för tillfället, men med underlag från exempelvis MODTRAN [11] kan även atmosfärspåverkan på lasersensorer simuleras.

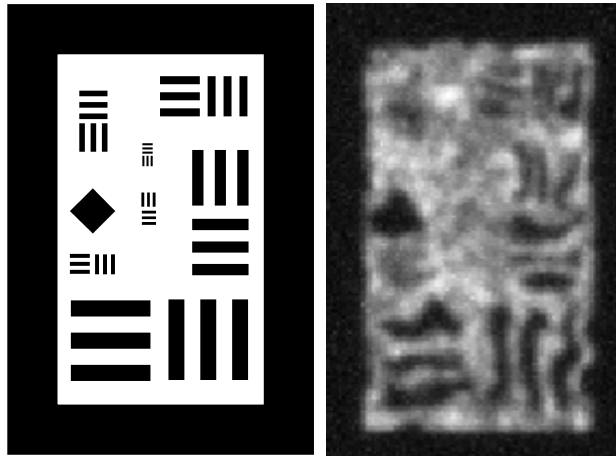
Avslutningsvis träffar de simulerade strålarna mottagarens optik och resulterar i en modellerad tidsupplöst elektrisk signal i detektorn. Denna fullständiga signal kan sedan degraderas för att simulera en specifik sensor. Vissa sensorer kan registrera vågformer, medan andra endast kan registrera avståndsbilder. *FOI-Ladarsim* är genom sitt upplägg flexibel och med rätt systemparametrar i simuleringarna, kan en stor variation av sensorer simuleras.

En simulering av hur ett 3D-avbildande lasersystem mot en stridsvagn T72 i småbruten terräng visas i figur 36. En simulerad reflektansbild visas till vänster och en simulerad avståndsbild visas i mitten. Till höger visas synfältet från sensorn i form av ett foto.



Figur 36. Till vänster visas en simulerad reflektansbild för en stridsvagn T72. I mitten visas en simulerad avståndsbild där ljusare bildpunkter är längre bort. Till höger visas ett foto som visar 3D-lasersensors synfält.

Under 2011 har fokus legat på uppdateringar i koden för turbulens- och atmosfärssimuleringar. Genom deltagande i NATO-gruppen "Modeling of Active EO-Imaging Systems" SET-156, har FOI tagit del av en förbättrad metod för simulering av strålgång genom atmosfär. Metoden liknar simulering med fasskärmar, en långsam metod där strålar propageras genom en volym med brytningsindexvariationer, men är enklare och snabbare [12]. Metoden tillåter regionala förskjutningar och utsmetningar av informationen såsom i figur 37, som visar en referenstavla som passerat genom turbulenssimuleringsmodulen i *FOI-Ladarsim*.



Figur 37. Referenstavlan till vänster har på bilden till höger passerat genom turbulenssimuleringsmodulen i FOI-Ladarsim utsatts för lokala förvrängningar på grund av den simulerade atmosfären. Simuleringen ska motsvara 1000 meters avstånd i medelstark turbulens.

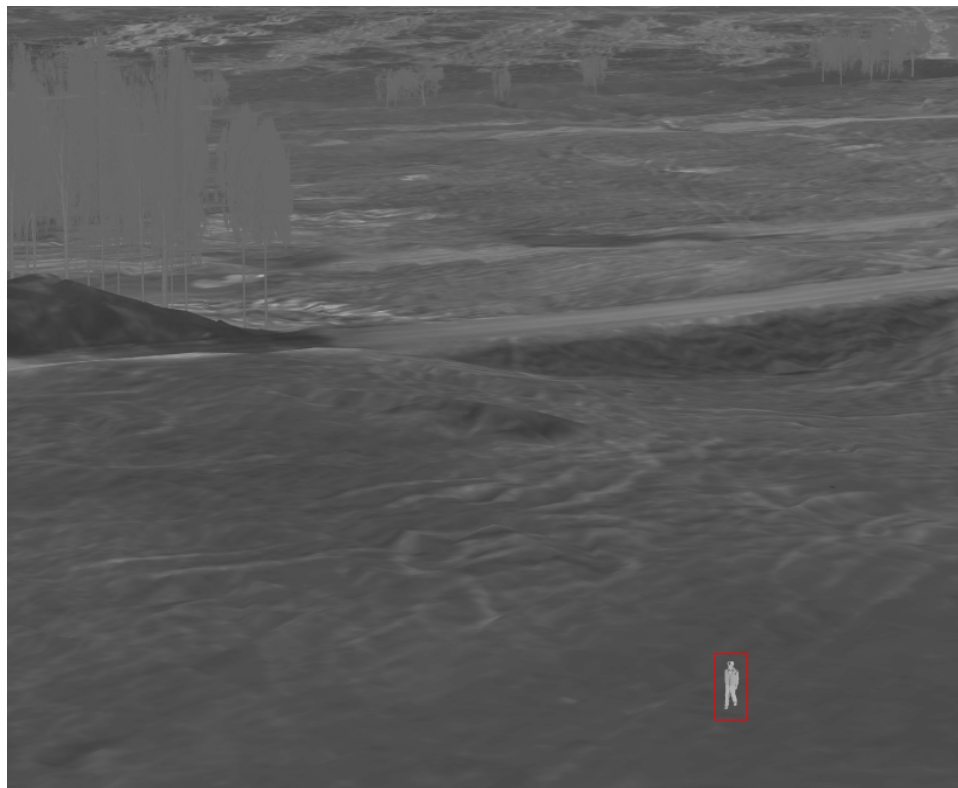
6 Resultat från värdering av spaning efter människor

I år har vi studerat förmågan för ett UAV-system att automatiskt spana efter människor. Ett exempel på person som vi vill kunna upptäcka visas i figur 38.



Figur 38. Exempel på person som vi vill spana efter.

Sensorn som vi har simulerat i denna studie är en IR-sensor som är känslig inom våglängdsbandet 8-9 μm . Simuleringarna har genomförts i Kvarnmodellen som är beskriven i kapitel 2.2. IR-signaturen har beräknats med RadThermIR och sedan använts som en textur på en animerad person. En bild från scenariot visas i figur 39 och i tabell 2 redovisas sensorspecifika parametrar. Vädret under simuleringarna var en mulen augustidag med temperaturen 17°C.



Figur 39. IR-simulering av en animerad person i Kvarnmodellen.

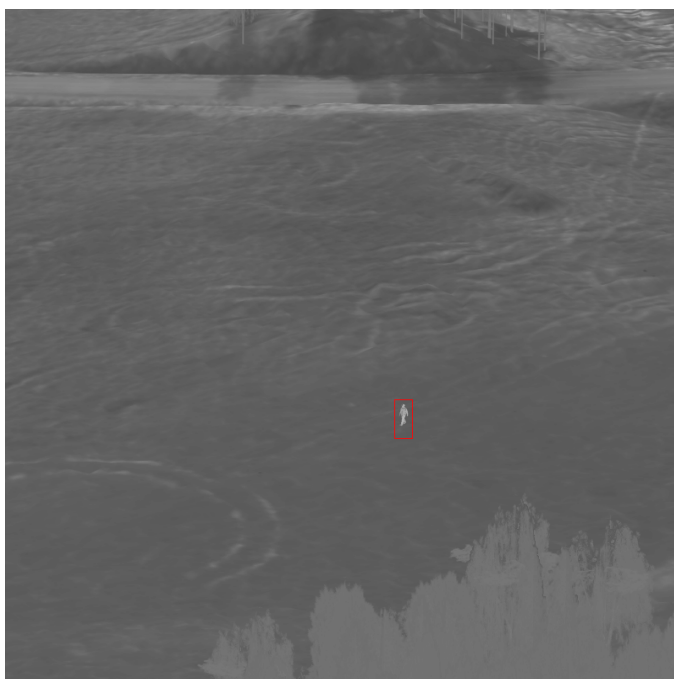
Tabell 2. Några data för IR-sensorn.

Avstånd	400 m
FOV	8×8 grader
Upplösning	800×800 pixlar
Våglängd	8-9 μm

6.1 Detektion

I projektet Signalbehandling för styrbara sensorsystem har en detektor/klassificerare tagits fram för att hitta människor i IR-bilder och denna har använts i denna studie för att detektera människor. Klassificeraren använder en variant av boosting som kallas LogitBoost [13]. Boosting bygger på att iterativt bygga en stark klassificerare genom att summera många svaga klassificerare [14]. En mer utförlig beskrivning av detektorn finns i [15].

Detektorn har tränats på riktiga IR-bilder från QWIP-systemet [15] för en januari-kväll. Detektorn är tränad på fyra olika skalor där personerna är 10, 15, 20 och 25 pixlar höga så fyra detektorer körs parallellt. Detta är gjort för att få både hög P_D (sannolikhet för detektion) och låg P_{Fa} (sannolikhet för falsklarm) för personer som är nära kameran (stora och lättare att träna en detektor för), samtidigt som detektorn för personer som är långt borta (små och svårare att träna en detektor för) kan ställas in till att ge låg P_{Fa} till priset av lite lägre P_D .



Figur 40. Simulerad IR-bild där en person går på ett fält i Kvarn. Personen detekteras automatiskt av UAV-systemet.

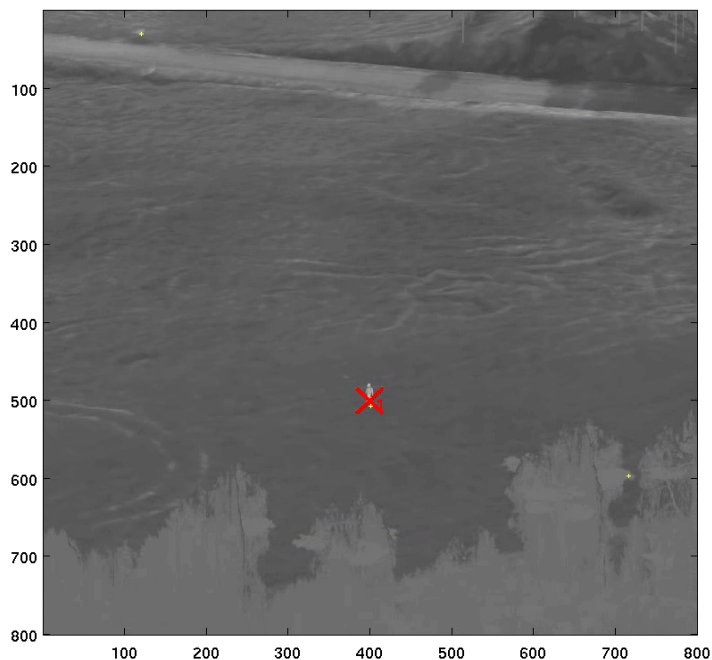
Detektorn är generellt mycket bra på att detektera människor och har låg sannolikhet för falsklarm. Andra objekt som djur och buskar detekteras vanligtvis inte. På IR-data med en upplösning av 320×240 pixlar kan människor detekteras med 36 bilder /sek om man tar hänsyn till tiden för inläsning av data från disk, och 50 bilder/sek om man

bortser från inläsning av data. Detektorn kördes på en dator med Intel Xeon CPU och 8 trådar användes vid försöket.

Trots att detektorn har tränats för ett helt annat väderfall än det som simulerats så detekteras personen. Om detektorn tränas om för rätt väderfall kommer troligen prestanda att kunna förbättras betydligt. På det här stadiet i projektet har ännu ingen formell evaluering av detektionsalgoritmen genomförts, men de erfarenheter som gjorts av detektionsalgoritmen är så här långt lovande.

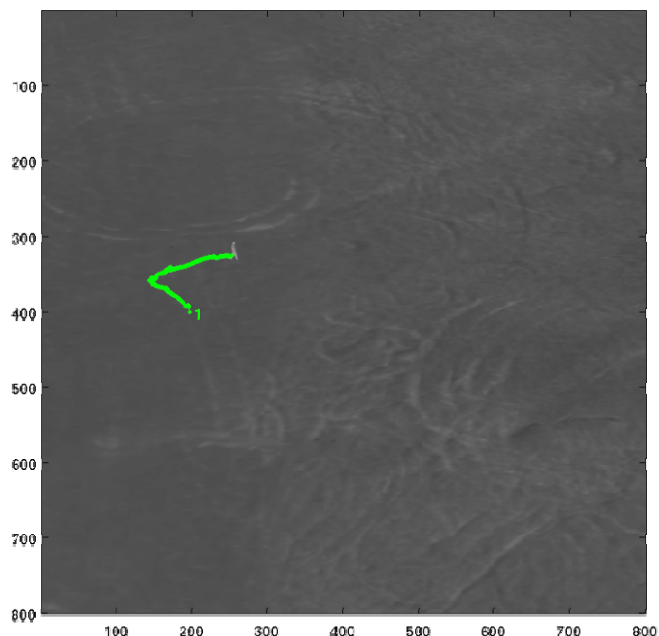
6.2 Målföljning

Målföljningsalgoritmen Multi-target Matlab Tracker [15] har använts för att skapa målspar från de givna detektionerna. Detta ger möjlighet att följa målens position över tiden, ge information om målpositioner även under begränsade tidsperioder utan direkta måldetektioner. Målföljningsalgoritmerna hjälper även till att stabilisera målsparen då måldetektionen ger oregelbundna resultat. Dessutom hjälper målföljningen till att sortera bort detektioner som inte beter sig som mål. På så vis kan antalet falsklarm minimeras. Figur 41 visar en simulerad IR-bild där målföljningen har överlagrats.



Figur 41. Resultatet från målföljaren har överlagrats på simuleringen. Målläget från målföljaren har markerats med ett rött kryss.

Trots att detektorn ibland registrerar falska detektioner så lyckas målföljaren att filtrera bort dessa, och endast det äkta målspar som följer personen blir kvar, vilket visas i figur 42. Målföljaren fungerar i realtid.



Figur 42. Ett målsår över tiden som visar hur personen har gått.

En målföljningsalgoritm utvärderas oftast med hänseende på fyra olika egenskaper; aktualitet, noggrannhet, kontinuitet och falska spår.

- Aktualitet – hur väl målföljningen kan adaptera till förändringar i scenen när nya mål dyker upp eller tidigare etablerade mål försvinner.
- Noggrannhet – hur väl målets position (och i förekommande fall hastighet och andra attribut) skattas.
- Kontinuitet – hur väl spår tillhörande ett specifikt mål hålls samman istället för att delas upp i olika spår potentiellt tillhörandes olika objekt, och hur ofta mål byter identitet med varandra under spårföljningen.
- Falska spår – hur väl målföljningen undertrycker spår som kommer från feldektioner.

På det här stadiet i projektet har ännu ingen formell evaluering av målföljningen genomförts, men de erfarenheter som gjorts av systemet så här lång är lovande givet det indata systemet haft tillgång till. I takt med att mer data blir tillgänglig kommer det också att bli möjligt att göra mer formella utvärderingar av målföljningsalgoritmens prestanda.

7 Fortsatt arbete

MSSLab har redan idag mycket bra funktionalitet, men simuleringsmiljön kan förbättras på många olika sätt då den snabba teknikutvecklingen inom simuleringsområdet och ökande datorprestanda hela tiden möjliggör ny funktionalitet.

För att få noggranna simuleringar av sensordata krävs att både terrängmodeller, materialklassificering, scensimulering och sensormodeller är validerade. Ett omfattande och viktigt arbete behövs för att validera alla simuleringsmodeller. Validering kommer i huvudsak att ske genom mätningar, men även med teoretiska beräkningar.

Tillgängliga omvärldsmodeller har god kvalitet vid simulering av luft till mark scenarier. Vid simulering av mark till mark scenarier är upplösningen allt för dålig i både Kvarn och Norrköpingsmodellen. Noggranna simuleringar av marina scenarier är inte heller möjliga med befintliga omvärldsmodeller, utan dessa måste i så fall vidareutvecklas.

En begränsning med MODTRAN är att molnen täcker hela himlen. Det går alltså inte att ändra molnighetsgraden. Ett sätt att lösa detta kan vara att generera egna modeller av moln och placera ut dessa som objekt i scenen som skall simuleras. En annan begränsning i MODTRAN är att det inte finns modeller för snöfall och hagel.

Atmosfärmodellerna för radar är mycket bra, men en noggrannare modellering av snöfall krävs för att få med inverkan vid högre frekvenser och för delvis smälta flingor, där det är känt att dämpningen kan överstiga den i regn, för samma ekvivalenta nederbördsmängd i smält form.

Fortsatt verksamhet med lasermodellen *FOI-Ladarsim* kommer att inkludera anpassningar till den gemensamma materialklassningen som gjorts för vissa scener, samt tillägg för att hantera ytterligare vädersituationer. Dessutom planeras valideringar av turbulensmodellen.

Väderpåverkan vid simuleringen av sensordata har begränsats till atmosfärseffekter. Förändringar av objektens och bakgrundens signatur för de olika väderfallen har således inte tagits med. Detta måste åtgärdas i det fortsatta arbetet.

8 Referenser

- [1] Näsström F., Allvar J., Berggren J., Carlsson L., Chevalier T., Haapalahti G., Habberstad H., Hedström J., Hemström F., Karlsson M., Karlsson N., Persson A., Rydell J., Skoglar P., Stenborg K-G., Tolt G., "Simuleringsbaserade metoder för sensorvärdering – Slutrapport", FOI-R--3118--SE, December 2010.
- [2] Stenumgaard P., Junholm L., "GENESIS - A tool for intersystem-interference analyses on camps", FOI-S--3105--SE, Linköping, 2009.
- [3] Internet: <http://www.turbosquid.com>, besökt 2009-12-03.
- [4] Bennet F. och Fenelius S., "SceneServer - a 3D software assisting developers of computer vision algorithms", FOI-R--0831--SE, Linköping, Mars 2003.
- [5] Internet: <http://www.openscenegraph.org>, besökt 2007-12-16.
- [6] Internet: <http://code.google.com/p/osgocean/>, besökt 2011-11-23.
- [7] Larson H., "Test plan for ASC FLASH 3D Ladar trials in Sweden, September 2008", FOI-D--0310--SE, Linköping, September 2008.
- [8] Chevalier T., "Statusrapport Modellerings av 3D-avbildande laser", FOI MEMO 2417, Linköping, Maj 2008.
- [9] Chevalier T., "Modeling of 3D laser radar systems. Status of FOI-LadarSim in May 2009", FOI-R--2764--SE, Linköping, Maj 2009.
- [10] Chevalier T., "A ray tracing based model for 3D ladar systems", pp 39-48, in Proceedings of the International Conference on Computer Graphics Theory and Applications, GRAPP, Vilamoura, Algarve, Portugal, March 5 - 7, 2011.
- [11] Kneizy F. X., Abreu L. W., Anderson G. P., Chetwynd J. H., Shettle E. P., Berk A. Bernstein L.S., Robertson D. C., Acharya P., Rothman L. S., Selby J. E. A., Gallery W. O., Clough S. A.: "The MODTRAN 2/3 Report and LOWTRAN 7 Model". Phillips Laboratory, Geophysics Directorate, PL/GPOS, Hanscom AFB, MA 01731-3010, USA . 1996 (Rapporten finns i PDF format på CD-ROM distributionen av PcMod-Win4.0.
- [12] Repasi, E. och Weiss R., "Computer simulation of image degradations by atmospheric turbulence for horizontal views," in SPIE Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII, vol. 8014. Orlando, FL, US: SPIE, 2011, pp. 8014-28.
- [13] Friedman J., Hastie T., Tibshirani R., "Additive Logistic Regression: A statistical view of boosting", The Annals of Statistics, 2000, Vol. 28, No. 2, pages 337–407.
- [14] Schapire R. and Singer Y., "Improving Boosting Algorithms Using Confidence Rated Predictions", Machine Learning, 37(3), 297-336, 1999.
- [15] Näsström F., Haapalahti G., Rydell J., Stenborg K-G., Skoglar P., Engström P., "Lägesrapport för projektet Signalbehandling för styrbara sensorsystem", FOI Memo 3611, 2011.