

FREDRIK NÄSSTRÖM, JAN GUSTAVSSON, GUSTAV HAAPALAHTI, JOHAN HEDSTRÖM,
FREDRIK HEMSTRÖM, MIKAEL KARLSSON, MIKAEL LUNDBERG, JONAS NILSSON, JONAS NYGÅRDS,
ANDREAS PERSSON, JOAKIM RYDELL, PER SKOGLAR, KARL-GÖRAN STENBORG, KATARINA ÅBERG



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, Jan Gustavsson, Gustav
Haapalahti, Johan Hedström, Fredrik Hemström,
Mikael Karlsson, Mikael Lundberg, Jonas Nilsson,
Jonas Nygårds, Andreas Persson, Joakim Rydell,
Per Skoglar, Karl-Göran Stenborg, Katarina Åberg

Simuleringsbaserade metoder för sensorvärdering - Årsrapport 2009

Titel	Simuleringsbaserade metoder för sensorvärdering – Årsrapport 2009
Title	Simulation based sensor system assessment – Annual report 2009
Rapportnr/Report no	FOI-R--2890--SE
Rapporttyp Report Type	Användarrapport
Sidor/Pages	43 p
Månad/Month	December
Utgivningsår/Year	2009
ISSN	ISSN 1650-1942
Kund/Customer	FM
Kompetenskloss	23 Värdering av telekrig och sensorsystem
Extra kompetenskloss	
Projektnr/Project no	E3097
Godkänd av/Approved by	Ola Kärvell
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

Projektet SIMSENS (Simuleringsbaserade metoder för sensorvärdering) har i uppgift att utveckla en gemensam simuleringsmiljö för värdering av sensorsystem i olika miljöer, väderförhållanden och tidpunkter. Simuleringsverktyget som vidareutvecklas för detta är MSSLab (MultiSensorSimuleringsLab).

För att en korrekt sensorvärdering skall erhållas med simuleringsbaserade metoder krävs noggranna simuleringsmodeller. Under året har SIMSENS-projektet förbättrat olika simuleringsmodeller för terräng, objekt och sensor. I denna rapport beskrivs arbetet som bedrivits inom projektet under 2009.

En stor del av årets verksamhet har varit att förbättra Norrköpingsmodellen och befolka den med människor och fordon. Fler byggnadstexturer har tagits fram och applicerats på Norrköpingsmodellen. Upplösningen på mark-texturen har ökat från 0.7 m till 0.3 m. De nya trädmodellerna har försetts med LOD (level of detail) så att de modelleras med hybridträd på nära avstånd och billboard-träd på långt avstånd.

För att göra det möjligt att simulera sensordata med animerade människor har HLAS, High Level Animation System, vidareutvecklats. En scenario-editor har tagits fram, vilket underlättar generering av scenarier med animerade människor.

Under året har laserfederaten uppdaterats och integrerats till MSSLab. SceneServer har vidareutvecklats med förbättrad ljussättning och skuggor för att de visuella simuleringarna ska se realistiska ut.

En simulering har genomförts med de nya simuleringsmodellerna för att studera sensorprestanda ur ett systemperspektiv i en urban miljö. Scenariot som har studerats består av två UAV-system som är utrustade med en visuell sensor respektive 3D-laser. Prestanda för detektion, målföljning, återigenkänning, ruttplanering och sensorstyrning har studerats för detta scenario.

Nyckelord: simulering, värdering, IR, laser, radar, MSSLab

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).

To get a correct sensor assessment with simulation based methods accurate simulation models are needed. During this year the SIMSENS' project has improved models for terrain, object and sensor. This report describes the work that has been performed in the project during 2009.

A lot of work has been performed to improve the Norrköping-model and populate it with people and vehicles. Several building textures have been developed and applied to the model. The resolution of the ground texture has been increased from 0.7m to 0.3m. The new tree models have been applied with LOD (level of detail) and modelled using hybrid trees at close range and billboard trees at long range.

To make it possible to simulate sensor data with animated persons, HLAS (High Level Animation System) has been further developed. A scenario editor has been developed that simplifies the creation of scenarios with animated persons. About 25 persons together with animals and vehicles are prepared to be used in the development of scenarios. These can of course be increased with new characters.

During this year the 3D laser federate have been updated and integrated to MSSLab. SceneServer has been further developed by improving lightening and shadows to get realistic visual simulations.

Simulations have been made with the new simulation models to study sensor performance in an urban environment. The scenarios that have been studied consist of two UAV systems equipped with a visual sensor and a laser system respectively. Performance for detection, tracking, reidentification, sensor control has been studied for this scenario.

Keywords: Simulation, assessment, IR, laser, radar, MSSLab

Innehållsförteckning

1	Inledning	7
2	Vidareutveckling av terrängmodeller	9
2.1	Norrköpingsmodellen	9
2.2	Kvarnmodellen	10
2.3	Afghanistanmodeller	11
3	Objektmodeller och animering av människor	13
3.1	Objektmodeller med visuelltextur	13
3.2	Beräkning av IR-signatur	14
3.3	Animering av människor	17
4	Simulering av sensorsystem	22
4.1	Laser	22
4.2	SceneServer	24
5	Resultat från värdering av sensorsystem i urban miljö	26
5.1	Scenario	26
5.2	Detektion	28
5.3	Målföljning	31
5.4	Återigenkänning	34
5.5	Ruttplanering och sensorstyrning	36
5.6	3D-laser.....	38
6	Diskussion	40
6.1	Nyttiggörande.....	40
6.2	Fortsatt arbete.....	40
7	Referenser	42

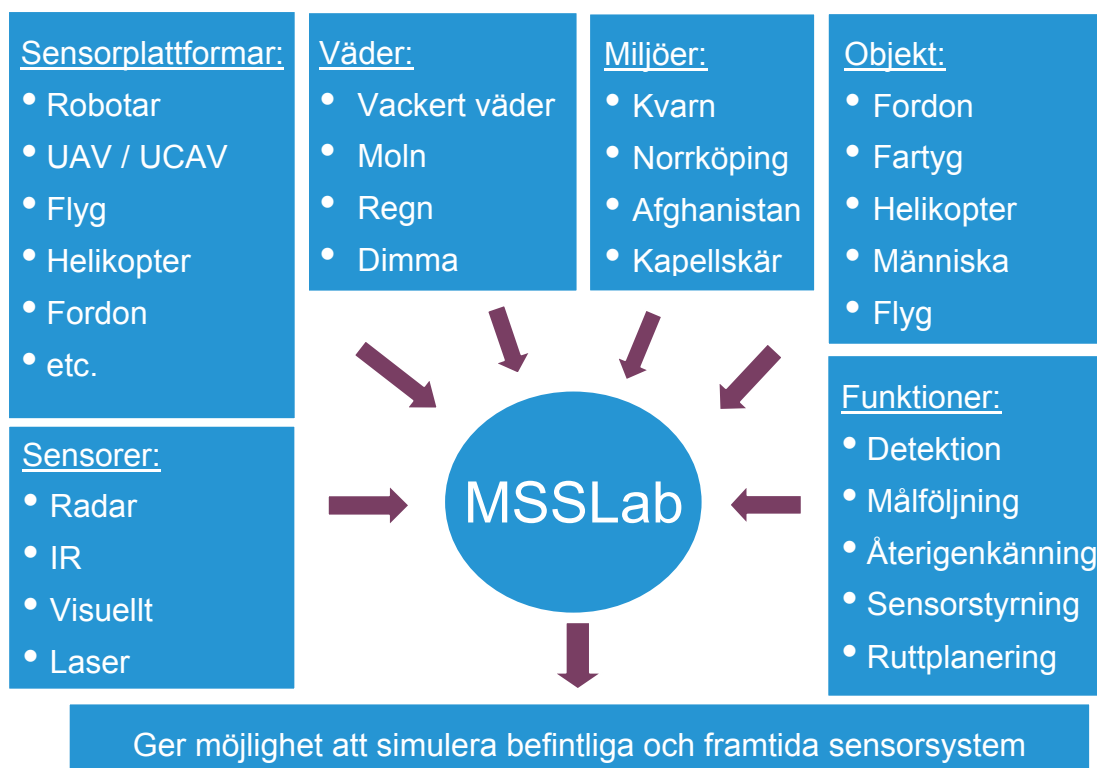
1 Inledning

Vid en 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, så kan simuleringar göras för en mängd olika fall till en rimlig kostnad och på relativt kort tid.

På FOI Informationssystem 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 (MultiSensor-SimuleringsLab), vilken är utvecklad vid FOI Informationssystem. MSSLab är en simuleringsmiljö där avancerade sensorsystem kan simuleras i olika miljöer och vid olika väderförhållanden och tidpunkter. Figur 1 visar olika beräkningsenheter 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 objektdektion, följning, återigenkänning, sensorstyrning och ruttplanering håller även på att integreras i 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: plats på jorden, årstid, dygnstid, lufttyp, luftfuktighet, temperatur, molnighet, molnbas, olika typer av nederbörd etc.
- Ljushöjdhållanden: tid på året och dygnet, aktuellt väder och plats på jorden.
- Datainsamling: observationstid, avstånd, höjd, aspektvinkelförändring på målet under observationen, hastighet på sensorplattformen etc.
- Hot och störning mot sensor och sensorbärare.
- Handlingsregler för uppdraget (ROE - Rules Of Engagement)

Projektet SIMSENS har inom vissa områden samarbetat med andra projekt under 2009. Exempel på detta är:

- Signalbehandlingsalgoritmer för detektion, målföljning, återigenkänning, sensorstyrning och ruttplanering som har tagits fram inom projektet Autonom spaning(ASP) och integrerats i samverkan med SIMSENS, ASP, ARCUS och Autonoma funktioner.
- Vidareutvecklat av SceneServer för att få realistiska visuella simuleringar med bra ljussättning och skuggor har skett i samarbete med SIMSENS och ASP.
- Vidareutveckling och integrering av laserfederaten till MSSLab har skett i samarbete med projekten SIMSENS, 3D-laserprojektet och SE-MARK.
- IR-modellering av HKP10 har skett i samarbete med projekten SIMSENS, Signaturmodellering och SE-MARK.
- En flygmodell för ett UAV-system har levererats av FoT-projektet Flexibla vapen och skydd.

2 Vidareutveckling av terrängmodeller

Under året har arbete utförts med vidareutveckling av terrängmodellerna för att göra det möjligt att simulera sensorsystem i olika miljöer. För närvarande finns följande terrängmodeller tillgängliga: Kvarn, Norrköping och Afghanistan.

Kvarn och Norrköping är geospecifika modeller vilket betyder att de är modellerade för att efterlikna verkligheten. Detta gör det möjligt att validera genomförda simuleringar med riktiga mätningar. Modellerna för Afghanistan är geotypiska modeller och dessa modeller är bra då man vill studera generella frågeställningar i sådana miljöer.

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 och modeller av enskilda träd inplacerade med korrekta dimensioner på korrekta positioner. Den rumsliga upplösning i modellen är ca 5-10 punkter/m² och med en koordinatnoggrannhet på 0.1 m (i rikets nät).

Norrköpingsmodellen har genomgått stora förbättringar under året. Fler byggnadstexturer har tagits fram och applicerats på modellen. De nya byggnadstexturerna används endast för att ge en realistisk variation av byggnaderna och de är inte applicerade efter byggnadstyp. Upplösningen på marktexturen har ökat från 0.7m till 0.3m. De nya trädmodellerna har försetts med LOD (level of detail) så att de modelleras med hybridträd på nära avstånd och billboard-träd på långt avstånd. Vid vidareutvecklingen har ett nytt verktyg som heter Trian3D [1] använts istället för Terravista som använts tidigare. I figur 2 visas en del av Norrköpingsmodellen som består av en högupplöst markyta, modeller av enskilda träd och över 25 000 byggnader.



Figur 2. Del av Norrköpingsmodellen.

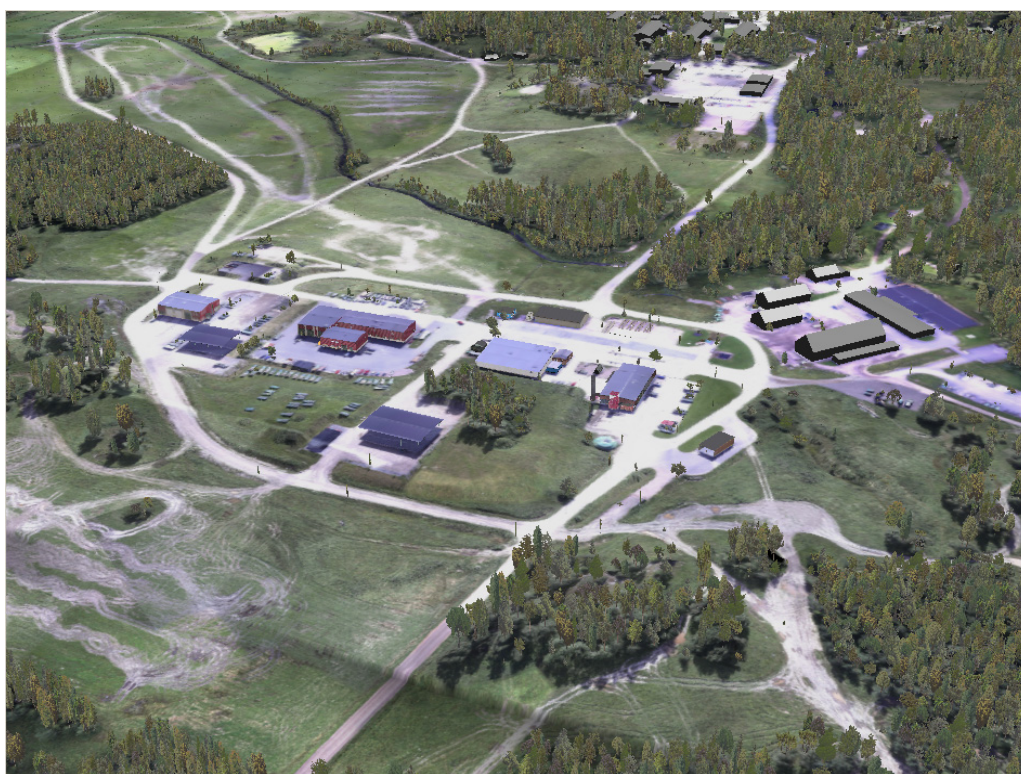
Information om vägnätet har använts för att lägga till vägar i modellen, vilket visas i figur 3. För taktextur hos modellen har ortorektifierat flygfoto använts. För att inte få ohanterliga mängder texturdata har husets takfärg skattats och används som textur. Detta möjliggör realistisk, om än bara delvis korrekt, visualisering av mycket stora områden.



Figur 3. Del av Norrköpingsmodellen. Husen är extraherade ur kommunens grunddata varefter generiska texturer applicerats på väggarna. Takfärgen har skattats från flygfoto.

2.2 Kvarnmodellen

Kvarnmodellen representerar en modell av Prästtomta skjutfält vid Kvarn som ligger ca 3 mil nordväst om Linköping. På samma sätt som för Norrköpingsmodellen så har detta område mätts upp med ett flygburet sensorsystem. Högupplöst visuellt och IR-data finns även insamlat över det laseruppmätta området. För Kvarnmodellen finns det även en materialklassad modell som används vid högkvalitativa sensorsimuleringar av IR- och radarsystem.



Figur 4. Bilden visar en del av Kvarnmodellen med uppmätt visuell textur.

2.3 Afghanistanmodeller

Det finns för närvarande två olika Afghanistanmodeller: Kabulmodellen och Camp GENESIS-modellen. Kabulmodellen består av ett bostadsområde i utkanten av Kabul och har en storlek på ca 500m × 500m. Information om markgeometri och husens placering har hämtats från Google Earth. Kabulmodellen består av ett ökenområde, fem rader med hus och ett antal träd och buskar.



Figur 5. Visuellt bild genererad med CameoSim.

Camp GENESIS är namnet på den modell som tagits fram i projektet "Nytt Telekonfliktverktyg" [2]. Modellen används i telekonfliktverktyget GENESIS som grund för ett typscenari där den svenska Campen i Maszar-e-Sharif varit inspirationskälla. Modellen är fiktiv och 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. I mitten av terrängen ligger själva förläggningen som en separat 3D-modell. Upplösningen på denna modell är högre än omkringliggande terräng då förläggningen är en central del i verktyget. Den omkringliggande terrängen kan där av betraktas som kuliss för själva förläggningen.



Figur 6. Camp GENESIS.

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 användas för att beräkna realistiska signaturer i olika scenarier.

3.1 Objektmodeller med visuelltextur

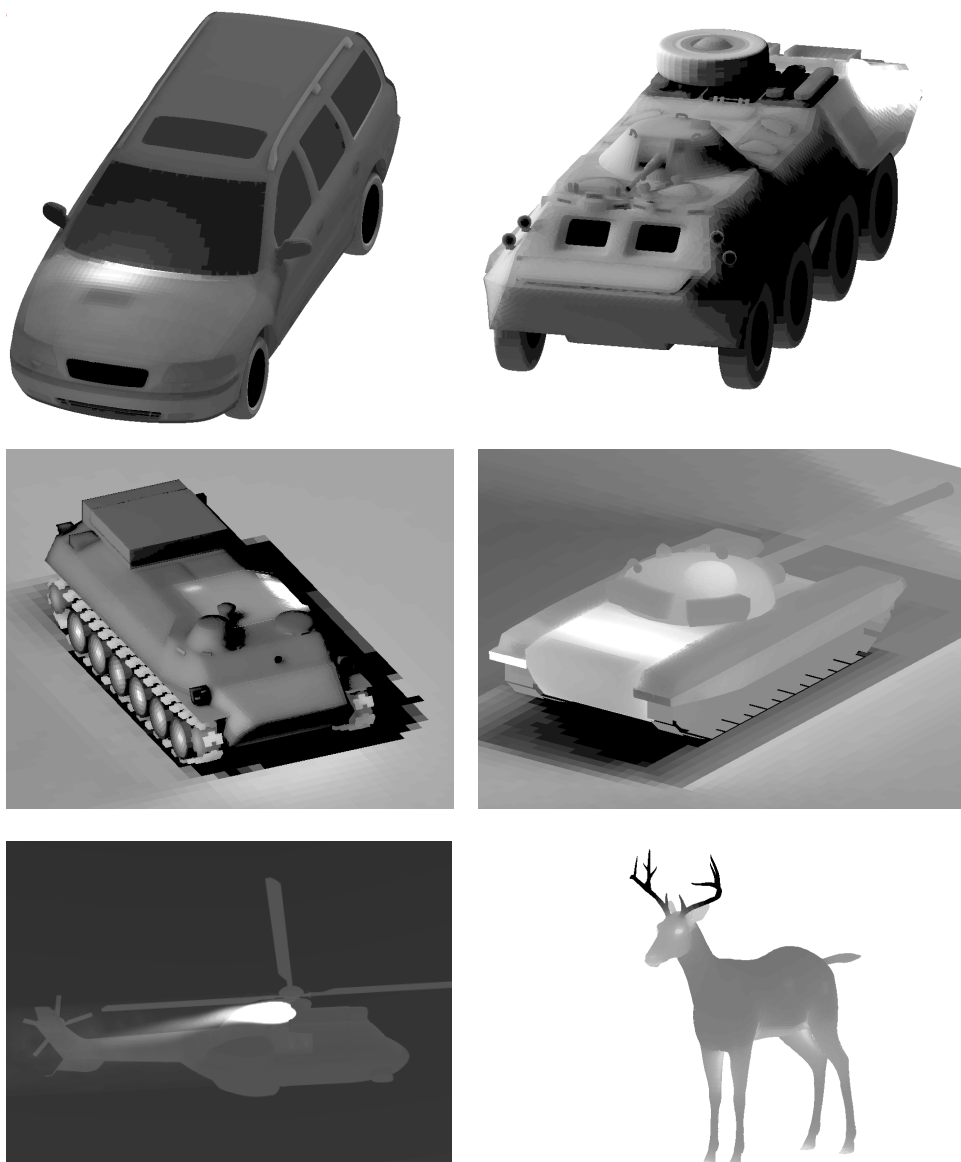
För scenariot som användes under året behövdes många olika objektmodeller. Några av de tillgängliga modellerna visas i figur 7 [3]. Arbetet med dessa modeller har i huvudsak bestått i att manuellt skala om dessa modeller och se till så att deras koordinatsystem är korrekt orienterat.



Figur 7. Några exempel på några tillgängliga objektmodeller med visuell textur.

3.2 Beräkning av IR-signatur

Arbete pågår inom SIMSENS-projektet för att beräkna IR-signaturer för olika objekt under skiftande drifts- och vädertillstånd. Fördelen med att beräkna IR-signaturen är att man kan analysera signaturen hos ett fordon med hänsyn till inre värmekällor, väder, vinddrag, friktionsvärme, tidpunkt på dygnet m.m. Analyser av uppvärmnings- och nedkylningsförlopp kan analyseras. Inre värmekällor kan till exempel vara motor, kylare och avgassystem. I figur 8 visas några resultat från signaturberäkningar.



Figur 8. Exempel på beräknade IR-signaturer. Från vänster till höger och uppifrån och ner: Volvo V70, BTR70, MTLB, T7, HKP10 och en hjort.

Under 2009 har arbete pågått för att modellera helikopter 10 (Super Puma). Slutmålet är att modellera alla till IR-signaturen viktiga bidrag, samt att senare också genomföra en validering mot mätdata. För detta arbete används programvarorna CFD++, RadThermIR och SE-RAY-IR, dessa kopplas till var-

andra för att därmed få med alla viktiga bidrag. Metodiken att beräkna IR-signaturer finns beskrivet i [4].

För en helikopter är alltid avgasplymerna en mycket viktig del av den totala IR-signaturen, även den konvektiva uppvärmning som sker på skrovet av avgasplymen kan vara betydande.

Här nedan visas aktuell status för arbetet. De första bilderna visar några strömningsbilder för helikopter 10 där den färdas i 220 km/h framåt och med randvillkor för avgasplymen gällande temperatur, gasblandning och massflöde. Resultaten från strömningsberäkningen är indata till RadThermIR, där man då även får med den konvektiva effekten på skrovet. Strömningsberäkningen är även indata till SE-RAY-IR för spektral plymrendering.

”Downwash” effekten från rotorn är beräknad genom en rotormodell i CFD++ där ett antal parametrar ges som randvillkor för att få med dess effekt. För strömningsberäkningen valdes k- ϵ turbulensmodellen. I figur 9 nedan visas hur randvillkor för rotormodellen sätts upp i CFD++.

Rotor Model information:

Activate the rotor model: yes no

The rotor model within CFD++ uses the parameters below to simulate the 3-D pressure rise across the rotor blades. The user should consult the CFD++ documentation for the definition of the parameters below. The rotor model can be applied both as a source term and a discontinuous jump across a pair of zonal boundaries.

File name for angle of attack data (cl, cd): Advanced Controls

Number of rotor blades:

Outer radius of rotor blades: m

Hub radius of rotor blades: m

Rotation speed of rotor blades (rev/min.):

Freestream velocity (excluding axial component): m/s

Collective pitch angle (theta0): degrees

Blade twist angle (theta1): degrees

Tilt of the tip path plane (a1, b1): degrees

Cyclic pitch parameters (A1, B1): degrees

Equilibrium angle (alpha_TPP): degrees

Coning of blades (a0): degrees

Fluid density (approximate): kg/m^3

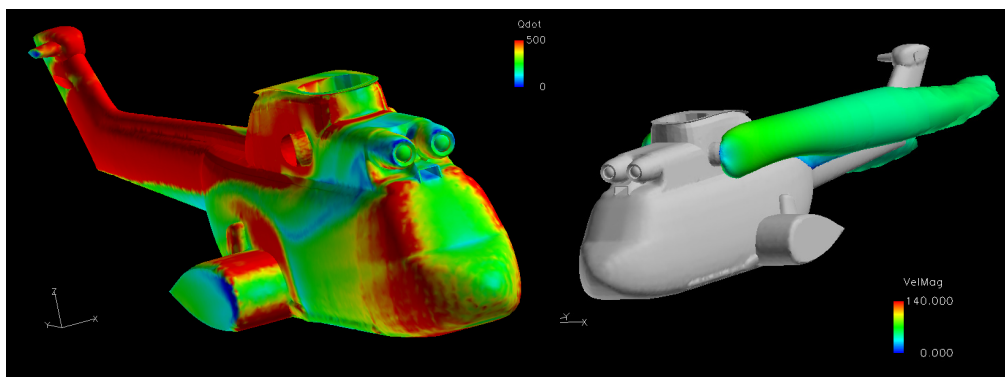
Accept and Exit Help Cancel and Remove InfoSet

Figur 9. Del av GUI i CFD++ där parametrar ges för rotormodellen, (inte hkp 10 värden).

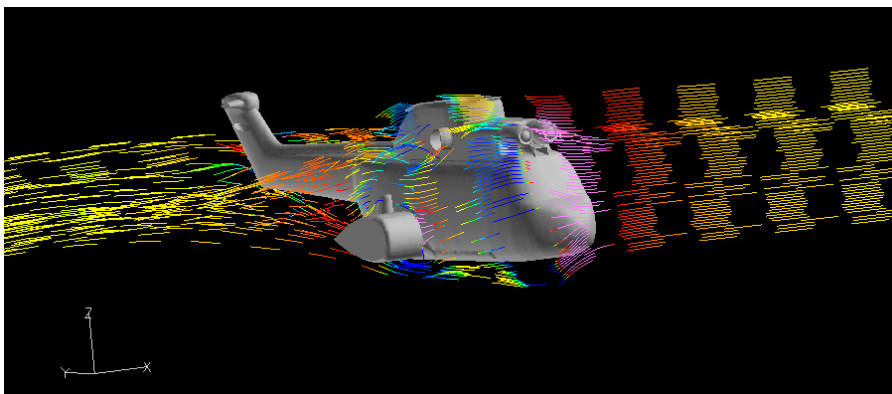
För extremt hög precision går det att göra beräkningen med rotorgeometrin och rörliga överlappande ”Chimera-meshar”. Fördelen med Chimera-meshar är att rotorn kan röra sig utan att det behöver skapas en ny mesh för varje tidssteg. För syftet med hkp 10 modelleringen är det dock tillräckligt att använda rotormodellen i CFD++. Denna rotormodell har validerats mot mätningar genomförda av Felker [5] där modellerad inducerad hastighetsvariation stämde mycket väl med uppmätt data.

I en hög hastighet som 220 km/h är inte effekten från huvudrotorn på avgasströmmen så uttalad, särskilt inte om man skulle jämföra med ett hovrande fall. Rotorn är ur meshsynpunkt ersatt med en platta där rotormodellen använts. I figur 10 nedan visas värmeffluxet (Q_{dot}) projicerat på skrovet, samt en isosvep-

ning av CO₂ i avgasplymen. Figur 11 visar en del av en animation där luftpartiklar färdas förbi helikoptern vid 220 km/h och den nedåtriktade luftströmmen från rotorn.

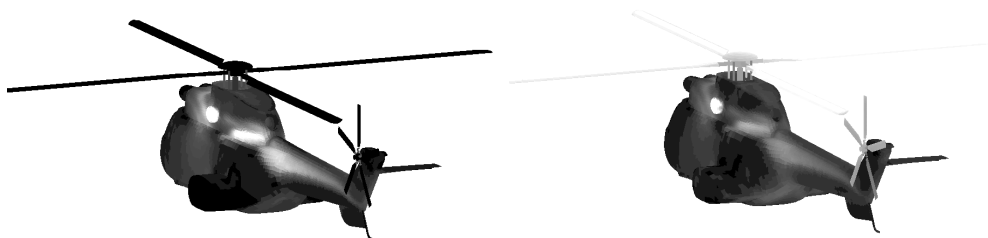


Figur 10. Till vänster visas värmeflux ($\dot{Q}$) beräknat i CFD++ och visualiserat i Fieldview för hkp10. Till höger visas CO₂ färgat med hastighetsmagnitud, beräknat i CFD++.



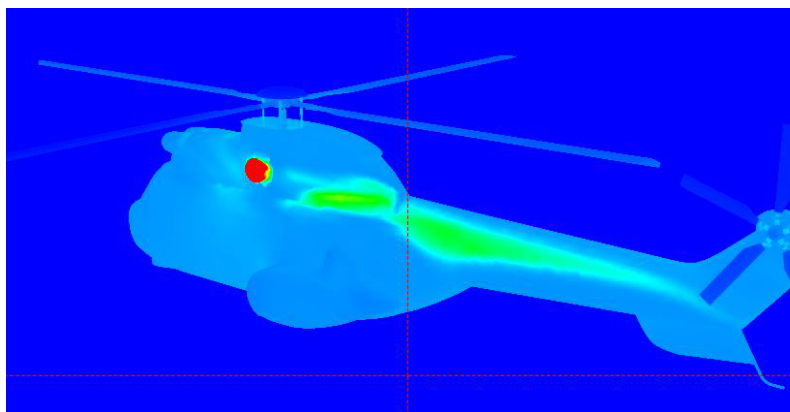
Figur 11. Flödeslinjer för helikopter då den färdas i 220 km/h framåt, beräknat i CFD++.

Modellering av hkp 10 har påbörjats i RadThermIR där resultat importerats från CFD++. Nedan ser man den konvektiva effekten på bommen, en dygnssimulering genomfördes transient var 5:e minut. Två tidpunkter visas nedan.



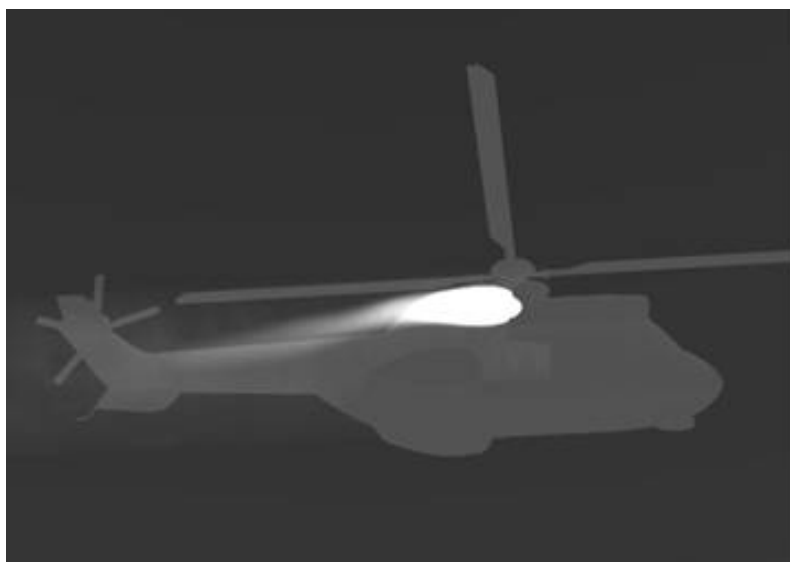
Figur 12. Hkp 10 modellerad i RadThermIR kl 00:00 till vänster och kl 12:00 till höger observera att det är olika temperaturskalor.

SE-RAY-IR kan importera resultat från RadThermIR, resultat visas i figur 13. Observera att det som syns i figur 13 endast är plymens konvektiva effekt på bommen och alltså inte själva plymen.



Figur 13. Resultat från SE-RAY-IR importerat från RadThermIR. Den konvektiva effekten orsakad av avgasplymen syns tydligt.

I figur 14 nedan visas avgasplymen renderad i MWIR (3-5 μ m) i SE-RAY-IR. Avgasplymen är skild från skrovet och sprids bakåt av fartvinden. Vid själva avgasutblåset ligger temperaturen på ca 500 C.



Figur 14. Plym renderad i SE-RAY-IR. Observera att RadThermIR resultat inte importerats här. Detta är en första "testrendering" och därmed inte är fysikaliskt korrekt.

Kommande arbete är att i SE-RAY-IR ta in både RadThermIR resultat och CFD++ resultat samtidigt, samt göra mer tester av plymrenderingen. Helikoptern kan därefter simuleras i olika miljöer och terrängar. I det fortsatta arbetet är det intressant att studera signaturen då helikoptern hoverar samt när helikoptern är utrustad med IR-suppresorer.

3.3 Animering av människor

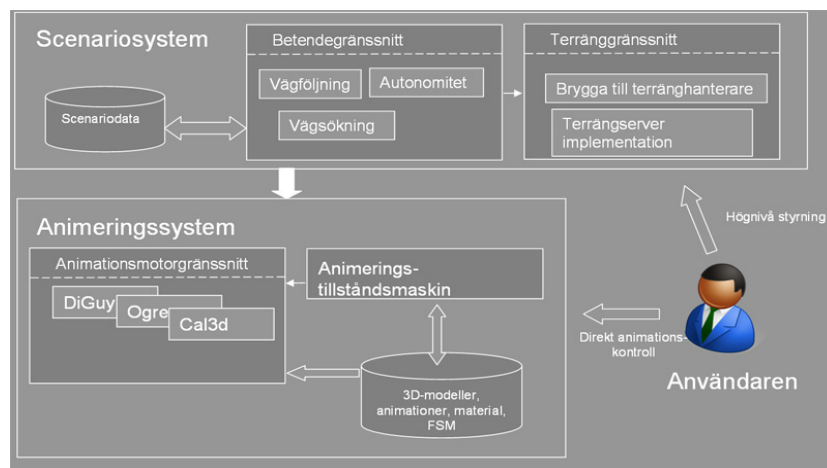
Simulering av sensordata med animerade människor är en verksamhet som påbörjades 2008. För att göra detta möjligt har HLAS, High Level Animation System, tagits fram inom SIMSENS-projektet. Med HLAS underlättas integra-

tionen av animerade sekvenser för olika typer av simuleringsramverk. För att skapa och visualisera skelettbaserade 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. I figur 15 visas några modeller som använts för animering av människor. Dessa är inköpta från [3].



Figur 15. Modeller som används vid animering av människor.

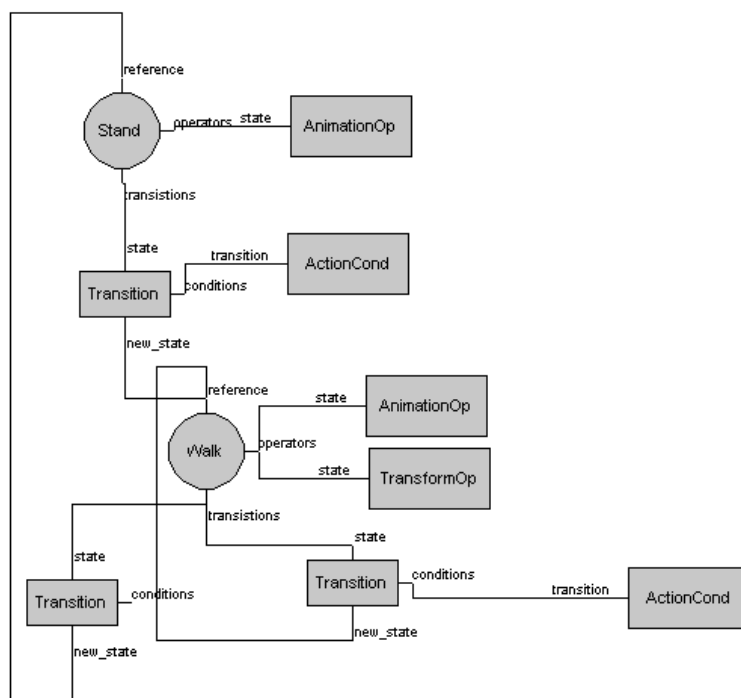
Grundtanken med designen av HLAS är att skapa ett plattforms- och renderingsberoende system som på ett enkelt sätt ska kunna utvidgas, integreras och anpassas för olika typer av simuleringar. HLAS kan delas in i två moduler, scenariosystemet samt animeringssystemet. Animeringssystemet kan i sin tur brytas ner i två delar, en tillståndsmaskinmodul och animationsmotormodul, vilket visas i figur 16.



Figur 16. Övergripande modulindelning.

Animationsmotorn är den del som hanterar inläsning, uppspelning och rendering av animationer. Då det redan existerar några system som hanterar just detta har denna del abstraherats via ett gränssnitt så att användaren har möjlighet att integrera valfria mjukvara på denna nivå. Referensimplementeringen av detta gränssnitt använder sig av animeringsmotorn Cal3D [6]. Eftersom Cal3D i sig är ett plattforms- och renderingsoberoende system lämpar det sig väl som referensimplementering. De renderingsimplementationer som i dagsläget finns till Cal3d är OpenGL [7], OpenSceneGraph [8] och Ogre3d[4]

En tillståndsmaskin i HLAS hanterar de animationer och förflyttningar en karaktär kan utföra. Tillståndsovergångar utförs då givna övergångsvillkor är uppfyllda. Exempel på ett sådant villkor kan vara ett kommando från användaren eller att karaktären uppnått en given hastighet. Då endast ett tillstånd kan vara aktivt vid samma tillfälle rekommenderas användaren att skapa separata tillståndsmaskiner för olika delar av karaktären. På detta sätt kan antalet tillstånd hållas ner till en hanterbar mängd. En tillståndsmaskin är konfigurerbar antingen via en xml-fil eller i ett grafiskt verktyg, vilket visas i figur 17.



Figur 17. Exempel på en tillståndsmaskin med två tillstånd, Stand och Walk. Dialogen visar parametrar för en animations operator.

För att få en flexibel tillståndsmaskin har ett tillstånd brutits ner i två delar, operators och transitions. En operator är något som påverkar karaktären t.ex. att en animation spelas eller att en förflyttning föreslås. Ett tillstånd kan ha godtyckligt många animationsoperatorer för att på så sätt skapa en önskad rörelse. Animationerna kommer vanligtvis från ett motion capture system, men kan även vara skapad i ett animeringsverktyg som t.ex. Motion Builder eller 3D-Studio Max.

En transition är en möjlig övergång till ett nytt tillstånd. En transition består av olika conditions vilka tillsammans avgör om en tillståndsovergång är möjlig.

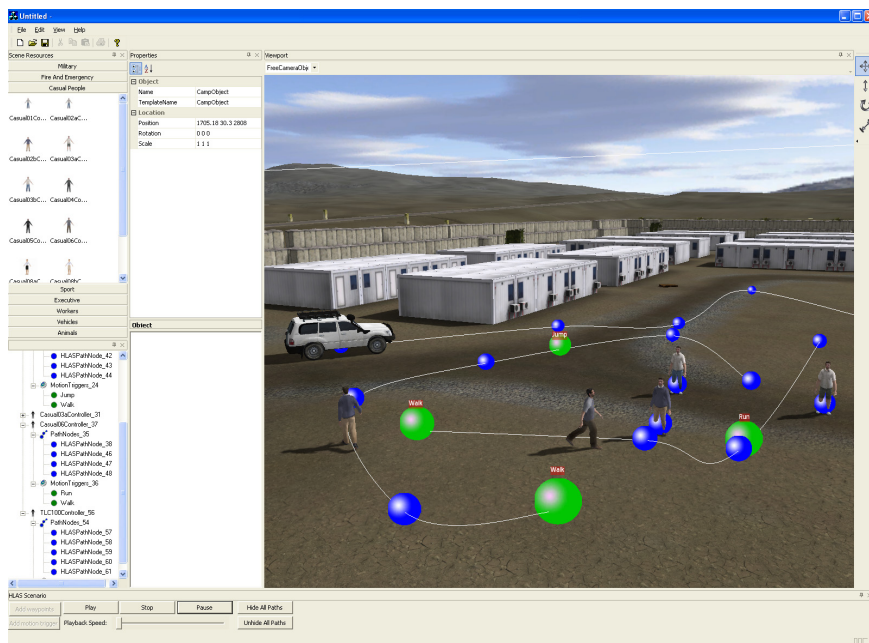
Existerar inga conditions för en transition är det en ovillkorlig tillståndsövergång. Om flera transitions är uppfyllda i samma tillstånd är det den först listade övergång som väljs, processordningen är med andra ord sekventiell. Om användaren vill utföra en slumpmässig övergång utförs detta genom att ytterligare möjliga tillstånd lista i samma transition.

Innan ett scenario kan skapas i HLAS måste varje deltagande karaktär ha en fungerande tillståndsmaskin som uppfyller önskade förflyttningar och rörelser. Med HLAS följer ett 25-tal olika människor samt några djur och fordon som är förberedda på detta sätt för att kunna användas i ett scenario. Användaren är givetvis fri att utöka med nya karaktärer eller modifiera befintliga.

Användaren skapar ett scenario antingen via HLAS API (Application Programming Interface) eller genom att ladda in ett färdigt scenario från en xml-fil. Därefter kan man via API:et ändra delar i scenariot för att sedan spara ner det till fil igen.

Beteendet som finns implementerat i HLAS är en enkel brytpunktsnavigering, vilket visas i figur 18. Denna navigering bygger på 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. Det finns även en möjlighet att avaktivera en motion trigger efter en uppdaterings cykel, detta är användbart då man har en tillståndsanimation som ej loopar.

När en motion trigger är aktiv skickas dess kommando till karaktären som sedan tolkas av tillståndsmaskinen som i sin tur föreslår en förflyttning. Brytpunktsnavigeringen utför sedan själva förflyttningen genom att fråga karaktären vilken hastighet den vill röra sig med och räknar därefter ut den totala tillryggalagda sträckan längs banan och placerar karaktären på denna position. Eventuella rotationer som tillståndsmaskinen föreslår tas inte hänsyn till av brytpunktsnavigeringen utan karaktären antas hela tiden vara roterad längs brytpunktsbanans riktning.



Figur 18. Scenario med brytpunktsnavigering.

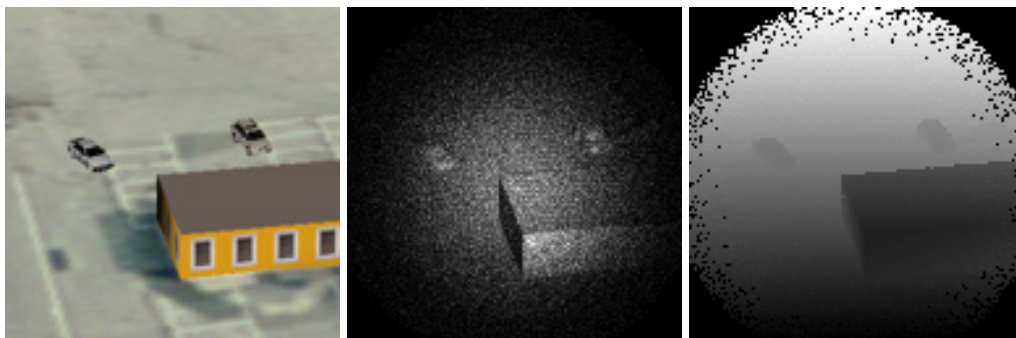
För att enkelt kunna skapa scenariofiler har ett scenarioverktyg utvecklats till HLAS. I scenarioverktyg kan användaren placera brytpunkter och motions triggers direkt i den omgivningen scenariot skall utspela sig i. Användaren har även möjlighet att spela upp, pausa och spola fram scenariot samtidigt som scenariot redigeras.

4 Simulering av sensorsystem

Arbetet med att simulera sensorsystem i Norrköpingsmodellen har kommit längst med 3D-lasermodellen och SceneServer. SceneServer har uppdaterats för att göra det möjligt att i nära realtid kunna simulera människor, djur och fordon i befintliga omvärldsmodeller. Båda modellerna är integrerade till MSSLab.

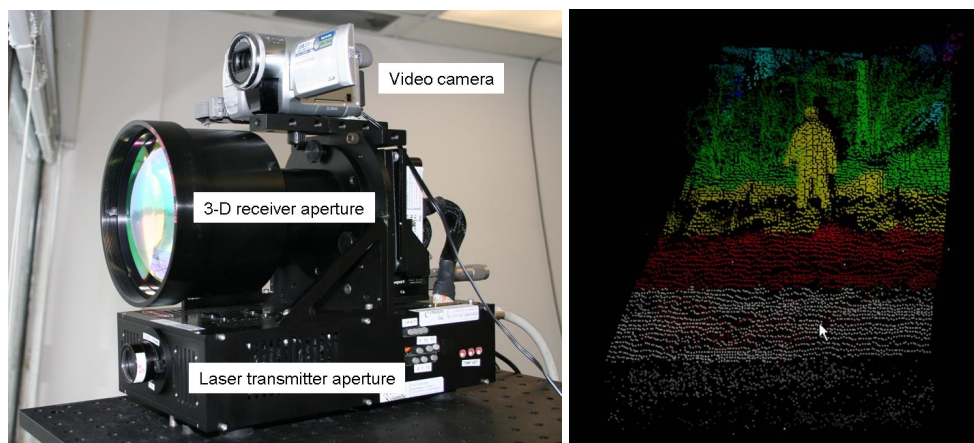
4.1 Laser

Årets arbete inom simulering av lasersystem har varit att integrera den befintliga lasermodellen till simuleringsmiljön MSSLab. Detta har varit ett samarbete mellan projekten SIMSENS, Måligenkänning med högupplösande 3D-avbildande laserradar och SEMARK. Tidigare utfördes simuleringar med lasermodellen helt med Matlab och SceneServer. Nu kan den simulerade lasern parametersättas och styras med hjälp av scenariemotorn NetScene. 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 19. Simuleringsresultat från ett scenario med två stycken bilar parkerade bredvid ett hus i Norrköping. Till vänster visas en bild av scenariot genererad av SceneServer. De två övriga bilderna visar simuleringsresultatet från laserradarn där bilden i mitten visar reflektans och bilden till höger avstånd.

Figur 19 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 [11][9] 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. Figur 20 visar en bild av Flash-sensorn och en avståndsbild av en människa i skogsmiljö inmätt med den verkliga sensorn. I tabell 1 redovisas sensorspecifika parametrar.



Figur 20. Vänster: Lasersensorn ASC Flash 3D. Höger: Avståndsbild som visar en människa i skogsmiljö.

Tabell 1. Laserparametrar

Våglängd	1,6 μm
Maximal pulsrepetitionsfrekvens (frame rate)	30 Hz
Sensorstorlek	128×128 pixlar
Pixelstorlek (pitch)	100 μm
Avstånd	3 – 1500 m
Avståndsnoggrannhet	± 7.6 cm
Synfält	9 – 0,5 °
Dimensioner (bredd x höjd x djup)	15 × 15 × 28 cm
Vikt	< 5,7 kg
Effekt	< 400 W
Beam Profile	Rectangular

3D-lasermodellen som har integrerats har utvecklats vid FOI under flera år. Programmet är i dagsläget baserat på SceneServer, vilket ger full insikt i koden och möjligheter att skraddarsy applikationer och anpassningar. Programmet bygger på en ray-trace-liknande princip och är uppdelat i fyra huvuddelar; laserkällan, scenen, atmosfären samt mottagaren [10,11].

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 används för att få med egenskaper mindre än ett bildelement.

För att generera ett tidsupplöst eko sänds simulerad laserenergi mot scenen i en stor mängd strålar. Det modellerade ekot genereras sedan genom att analysera hur varje stråle påverkas på sin färd mot 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 optisk energi färdas genom atmosfären påverkas den av turbulens och partiklar i luften. Atmosfäreffekterna 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 vädrets påverkan på lasersystem saknas för tillfället, men med underlag från exempelvis MODTRAN kan även atmosfärspåverkan på lasersensorer simuleras.

Avslutningsvis träffar de simulerade strålarna mottagarens optik och resulterar i en modellerad tidsupplöst elektrisk signal i detektorn. Genom parametrar som beskriver mottagaren, bestäms degraderingen av signalen. Hur den slutliga signalen ser ut bestäms genom systemdefinitionen. Vissa system resulterar i en vågform, medan andra endast ger ett eller flera avstånd i varje bildelement. Modellen tillåter godtyckliga system.

En simulering av ett 3D-avbildande lasersystem mot en stridsvagn T72 i småbruten terräng visas i figur 21. 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 21. 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-lasersensorns synfält.

4.2 SceneServer

Förenklade simuleringar av visuella och IR-sensorer kan göras med SceneServer. SceneServer [12] 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 skraddarsy applikationer och anpassningar. SceneServer är baserat på OpenSceneGraph [13] som är ett scenegraf-API för OpenGL.

SceneServer simulerar modeller baserade på 3D-geometri och texturer. Modellerna är uppbyggda av data från samtida insamlingar av laser, IR- och visuella data. Detta ger en mycket god representation av hur terrängen såg ut vid datainsamlingstillfället, 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.

Integrationen av SceneServer till MSSLab var tidigare gjord i Matlab, vilket gjorde det onödigt långsamt att simulera enklare optiska sensorsystem. För att snabba upp och förenkla dessa simuleringar har SceneServerfederaten under 2009 återimplementerats i JAVA. Implementationen har i stor utsträckning

följt Matlab-versionen för att så små anpassningar som möjligt ska behövas för de övriga redan existerande federaterna.

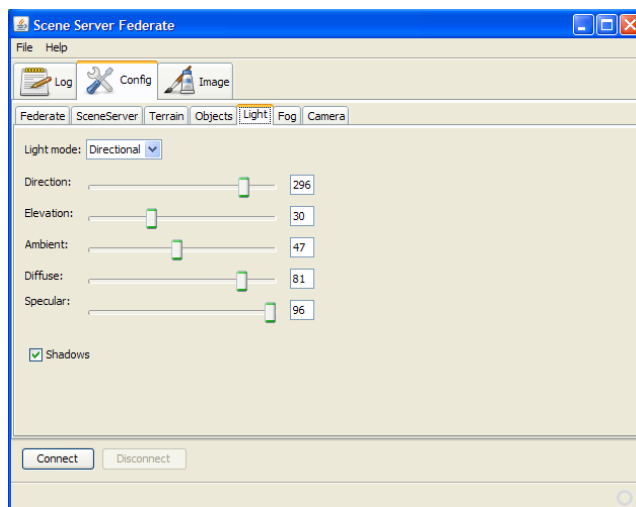
Under 2009 har animationsverktyget HLAS integrerats till SceneServer. Detta gör det möjligt att simulera animerade människor i befintliga omvärldsmodeller, vilket visas i figur 22. Scenarier från HLAS kan läsas in av SceneServer, vilket gör det möjligt att simulera stora grupper av människor.

För att de visuella simuleringarna ska se realistiska ut är det viktigt med bra ljussättning och skuggor. En mängd olika skuggtekniker har utvärderats. Endast ett fåtal har visat sig lämpliga för simulering av större 3D-miljöer med hundratals byggnader och träd. För att klara detta har skuggtekniken "Light Space Perspective Shadow Maps" [14] visat sig vara mest lämplig för SceneServer.



Figur 22. Simulering av människor och fordon med SceneServer.

I den nya SceneServerfederaten har ett antal nya funktioner införts. Dessa innefattar bland annat konfiguration av HLAS animering, konfiguration av ljus och skuggsättning samt laddning av terrängmodeller genererade i Trian3D. Ett grafiskt användargränssnitt har designats för federaten, vilket visas i figur 23. Detta förenklar användandet och framför allt konfigurationen av federaten. Konfigurationen sparas lokalt på den dator som federaten kör på.



Figur 23. Användargränssnittet för SceneServerfederaten.

5 Resultat från värdering av sensorsystem i urban miljö

Spaning i urban miljö är generellt svårt då byggnader ofta skymmer målområdet. En urban miljö även svår då den ofta innehåller många fordon och människor. I detta kapitel har vi simulerat två UAV-system för att värdera deras sensorprestanda i en urban miljö. Sensorerna som har studerats är visuellt och 3D-laser. För den visuella sensorn har dess förmåga till detektion, målföljning, återigenkänning, ruttplanering och sensorstyrning studerats.

5.1 Scenario

Ett scenario har använts för att studera prestanda för två olika sensorsystem. Bakgrunden till scenariot är hämtad från Bogalandscenariot, som bland annat har använts vid den gemensamma stabsövningen GSÖ 07. En målsättning vid framtagningen av Bogalandscenariot var att få med många av de olika företeelser och kombinationer av företeelser som förekommer i verkliga konflikter.

I Bogalandscenariot finns det två viktiga storylines som handlar om narkotika respektive bilbomber. Uppgiften är att kartlägga dessa verksamheter i Bogaland och inhämta underrättelser för att kunna lagföra individer som ägnat sig åt illegal verksamhet[15].

Inom ramen för scenariot har vi simulerat två MALE UAV-system (Medium-Altitude Long-Endurance Unmanned Aerial Vehicle). Dessa har förmåga att spana under lång tid över ett målområde, vilket är viktigt för att man ska kunna lösa de givna uppgifterna. Sensorerna som har studerats är visuellt och 3D Laser. Data för de olika UAV-systemen visas i tabell 2.

Tabell 2. Data för UAV-systemen.

UAV - Visuellt	UAV – 3D Laser
Speed: 40 m/s - 144 km/h	Speed: 40 m/s - 144 km/h
Altitude: 600 meter	Altitude: 150 meter
FOV: 10×10 grader	FOV: 0.1×0.1 grader
Resolution: 1000×1000 pixlar	Resolution: 128×128 pixlar
Framerate: 25 Hz	Framerate: 30 Hz
Wavelength: 0.38-0.75 μm	Wavelength: 1.57 μm

Scenariot som har simulerats inleds med att ett misstänkt fordon har upptäckts i centrala Norrköping, vilket visas i figur 24. Information om fordonets modell, färg, position och tidpunkt skickas till UAV-systemen som ska samla in information om dess rörelser. Denna information kan sedan användas för att ge en bättre bild av omfattning, medel och metoder hos den illegala verksamheten.



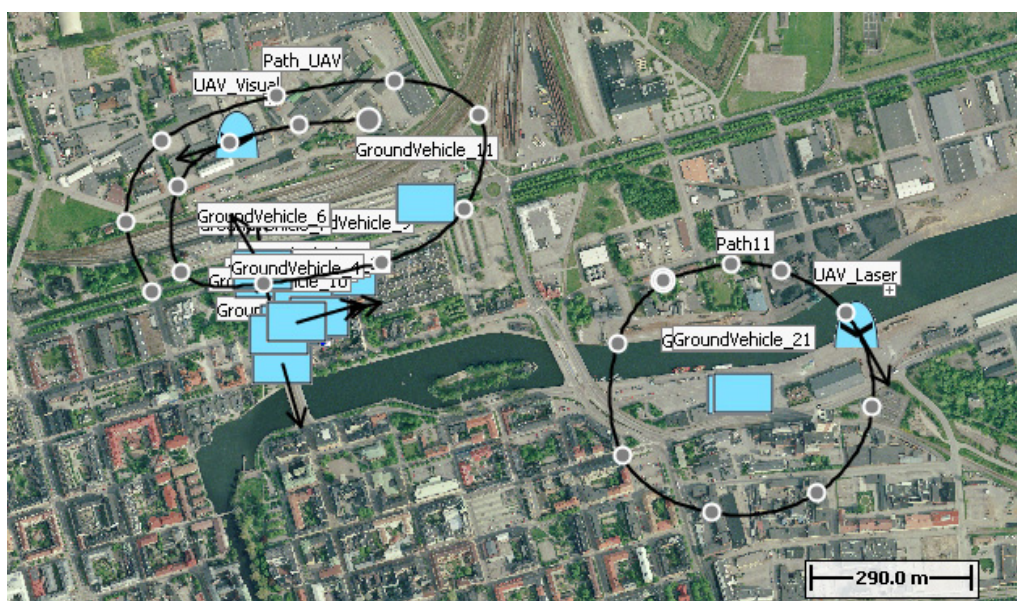
Figur 24. Två misstänkta fordon i scenariet.

Då UAV-systemet får information om fordonets modell och färg kan den använda sig av cad-modeller i ett målbibliotek för att träna sina algoritmer för att sedan kunna leta efter liknande fordon. Cad-modeller för de misstänkta fordonen visas i figur 25.



Figur 25. Cad-modeller för de misstänkta fordonen.

Då UAV-systemet närmar sig målområdet börjar det att detektera, målfölja och återigenkänna alla fordon i området. Figur 26 visar flygbanorna för de två UAV-systemen och fordonen i scenariot. UAV-systemet till vänster i bilden är utrustad med en visuell sensor och UAV-systemet till höger i bilden är utrustad med en 3D-laser.



Figur 26. Beskrivning av flygbanorna för de två UAV-systemen och fordonen i scenariot.

5.2 Detektion

En detektor har tagits fram för att hitta fordon i de visuella bilderna. Detektorn som använts är baserad på algoritmer som utvecklas i projektet Autonom spanning och mer information om detaljerna finns i referensrapporterna [16]. Träningen av algoritmen har utförts med hjälp av syntetiska data som genererats med SceneServer och Norrköpingsmodellen.

Tre olika 3D-bilmodeller, Volvo V70, Toyota Rav4 och BMW E36 har använts, och dessa har färglagts med olika lackfärg för att få med en sådan variation. För alla kombinationer av bilar och färg så har bilder genererats då modellen placerats ut i Norrköpingsmodellen. Bilarna placerades mestadels på vägar men även på parkeringsplatser, gräsytor och liknande. Bilder har genererats för alla infallsvinklar och orientering för bilarna. För varje vy har bilen placerats på 5 olika platser för att få variation på bakgrunden. Ljussättningen har varierats slumpmässigt för att inte detektorn ska tränas mot en viss ljusriktning. I detektorn som har tränats fram användes inte skuggor, men detta bör åtgärdas i framtiden.

Detektorn är uppdelad i en hierarki av kaskader där den första kaskaden är tränad för alla bilar och vyintervall. Den första kaskaden använder bara rektangelsärdrag [17] och är ganska dålig på att förkasta bakgrunder. Den kan dock ta bort en stor del av bakgrundsbilderna, vilket underlättar för de följande kaskaderna. Efter den första kaskaden är detektorn uppdelad i 8 kaskader som är specialiserade på en viss del av vyintervallet, vilket gör dom mycket bättre på att separera mål från bakgrund. I de 8 vyspecifika kaskaderna har först rektangelsärdrag använts och senare i kaskaderna har färgsärdrag baserade på SURF(Speeded up Robust Features) särdraget använts. I figur 27 visas ett exempel på resultat från den färdiga detektorn.

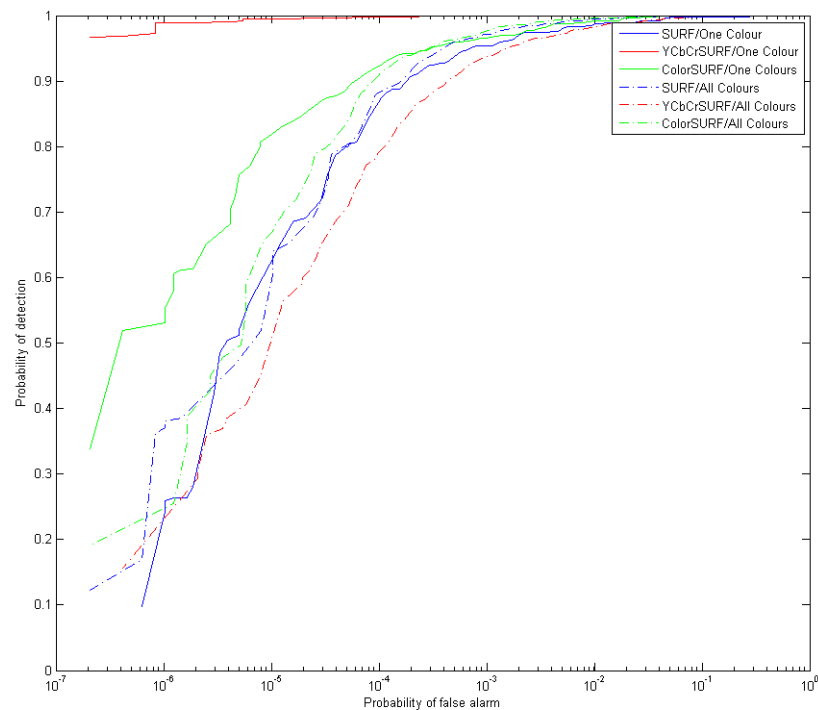


Figur 27. Detektioner med visuella data i Norrköpingsmodellen.

Nya färgsärdrag har använts av detektorn och prestanda för dessa har studerats. För att kunna studera prestanda har en kaskad med 20 komponenter tränats fram för varje typ av särdrag. Separata kaskader har gjorts för fordon med alla typer av färger samt en där vi bara använder röd färg på fordonen. Dessa kaskader används sedan för att generera ROC-kurvor (Receiver Operating Characteristic). Ett vyintervall på 45 grader ($1/8$ varv) används för att motsvara de djupaste delarna av hierarkin som använts i hela detektorn. Data som använts för träning är inte samma som används vid kurvgenereringen.

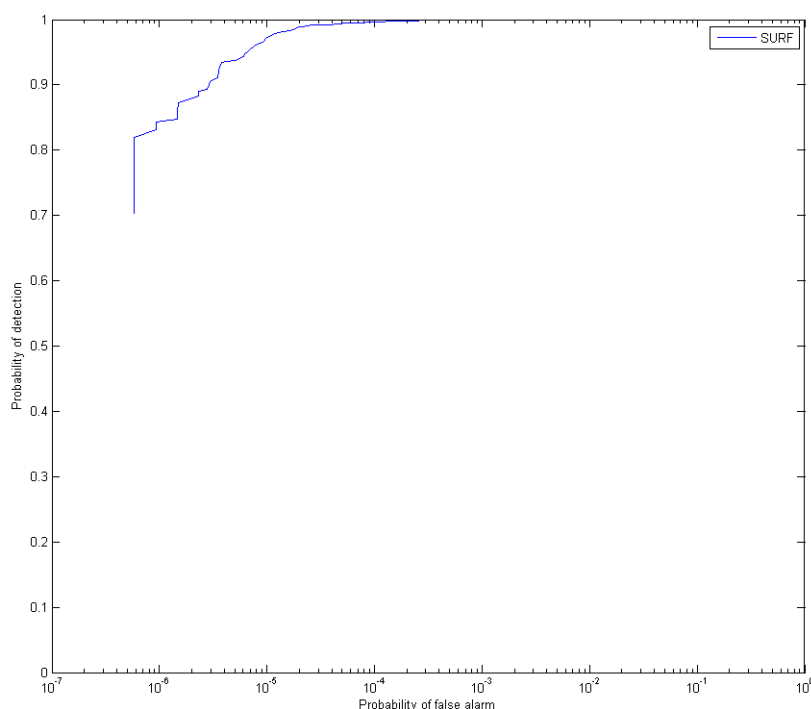
De särdrag som används är de som vi kallar SURF, ColorSURF och YCbCr-SURF. SURF jobbar endast på gråskalebilder, ColorSURF bygger på att varje pixel får 11 sannolikhetsvärden som beskriver hur stor chans det är att den tillhör en viss färgklass. Dessa summeras sedan vektorvis i SURF regionerna (4 i vårt fall) och konkateneras till särdragsvektorn. YCbCrSURF är en beräkningsmässigt enklare variant av ColorSURF som konverterar varje pixel från RGB till YCbCr representation. På Y-kanalen appliceras ett vanligt SURF-särdrag och på Cb och Cr kanalerna summeras pixelvärdena i SURF regionerna och konkateneras till särdragsvektorn.

Prestanda för de olika särdragen visas i figur 28. Resultaten visar att om flera olika färger används på bilarna så blir vinsten relativt liten då färgsärdrag används, men om bara en färg används så ger det desto mer. I ett fortsatt arbete är det intressant att undersöka prestanda för en detektor där en hierarki tränas med separata detektorer för varje färg, på samma sätt som idag sker med vyintervall [16]. Sannolikt ger detta bättre resultat.



Figur 28. ROC-kurvor för de olika särdragen och de olika träningsdata/evalueringsdata som används.

För att ha något att jämföra den visuella detektorn med så har prestanda för en IR-detektor tagits fram, vilket visas i figur 29. Detektorn är tränad för alla rotationsvinklar som fordonen kan ha till skillnad från den visuella detektorn där prestanda togs fram för ett begränsat rotationsintervall ($1/8$ varv). Trots att IR-detektorn jobbar på ett större vyintervall är prestanda bättre än för den visuella detektorn. Prestandaskillnaden beror förmodligen på att fordonen i IR-fallet simuleras som varmkörda, vilket ger en tydlig kontrast mot bakgrunden.



Figur 29. ROC-kurva för en IR-detektor som använder lika många komponenter som den visuella detektorn.

5.3 Målföljning

Från detektionsalgoritmen kommer detektioner av mål (fordon) för varje bild. Målföljaren associerar sedan dessa observationer över tiden till målspar som ger möjlighet att följa och prediktera målens position och hastighet över tiden.

Målföljaren som har använts för denna studie är mycket generellt uppbyggd och för tillfället finns det moduler för Kalman filter (”Extended” och ”Unscented”) och partikelfilter för tillståndsestimeringen [18]. För association av observationer till spår finns två moduler GNN (Global Nearest Neighbour) och JPDA (Joint Probabilistic Data Association). I denna studie användes ett ”Extended” Kalmanfilter för tillståndsestimeringen och JPDA för association av observationer till spår.

Figur 30 visar en bildsekvens från scenariot där fyra bilar färdas i en kolonn. En femte bil, det fordon som har målsparidentitet 85 i de tre bilderna, står parkerad i det högra körfältet och måste därför köras om av bilarna i kolonnen. Multimålföljaren bibehåller identiteterna på fordonen med målsparidentitet 85 och 103 under omkörningen, vilket tyder på lyckad tillståndsuppskattning samt lyckad association av observation till spår.

I den andra bilden i sekvensen har ett nytt målspar med identitetsnummer 118 uppstått på grund av bakgrundsklotter. Få detektioner genereras i närheten av det nya målsparret, vilket innebär att målsparrets poäng minskar över tiden. När spårrets poäng blir för låg raderas spåret, vilket kan ses i den tredje bilden i sekvensen.



Figur 30. Bildsekvens från simulerat scenario i Norrköping: fyra fordon färdas i kolonn och måste köra om den parkerade bilen som i bilderna indikeras av målsparidentitet 85.

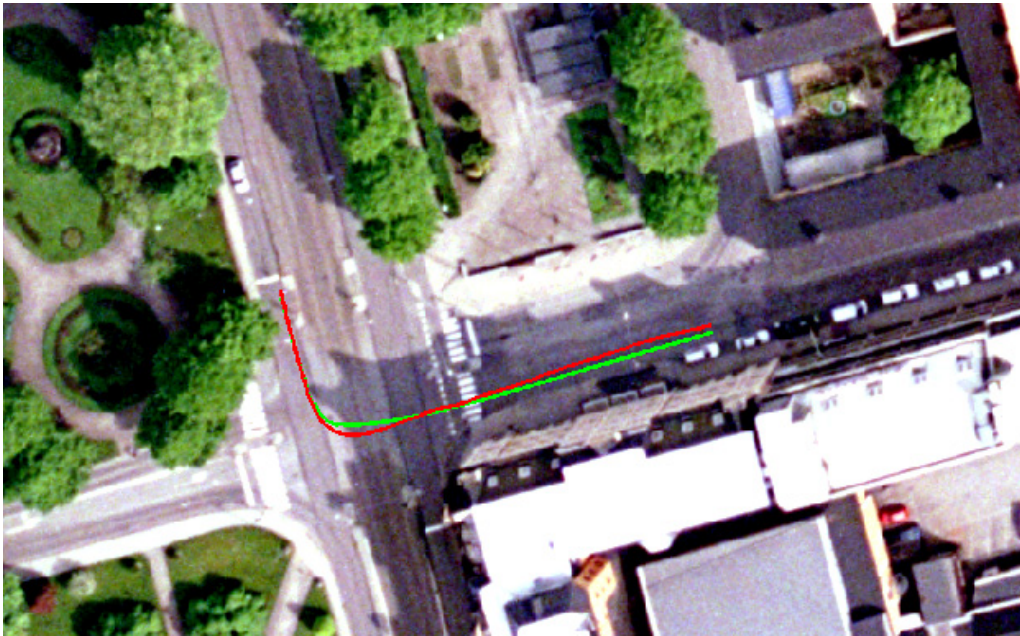
I bildsekvensen som visas saknas detektioner för det sista fordonet i kolonnen mellan den andra och den tredje bilden. Eftersom detektioner saknas minskas poängen för detta spår och när poängen blivit för låg raderas målspar. Då fordonet detekteras igen kommer således ett nytt målspar med ett nytt identitetsnummer att uppstå, i det här fallet identitetsnummer 123. Detta är ett exempel på när återigenkänningsmodulen kan användas för att bibehålla identiteten på fordonet.

Då ett mål inte detekteras under en tid kommer osäkerheten gällande dess tillstånd att öka. I figur 31 visas en bildsekvens där ruttplaneraren använts. Under sekvensen styrs sensorn över vägen där fordonen färdas. I den första bilden har den gröna bilen inte detekterats ännu. Då fordonet inte detekteras under tiden som fordonet befunnit sig utanför sensorns synfält har osäkerheten gällande dess tillstånd ökat. Detta illustreras av cirklarna som omger prediktionerna. I den andra bilden, som kommer från en tidpunkt något senare, har den gröna bilen detekterats vilket därmed minskat osäkerheten gällande dess tillstånd.



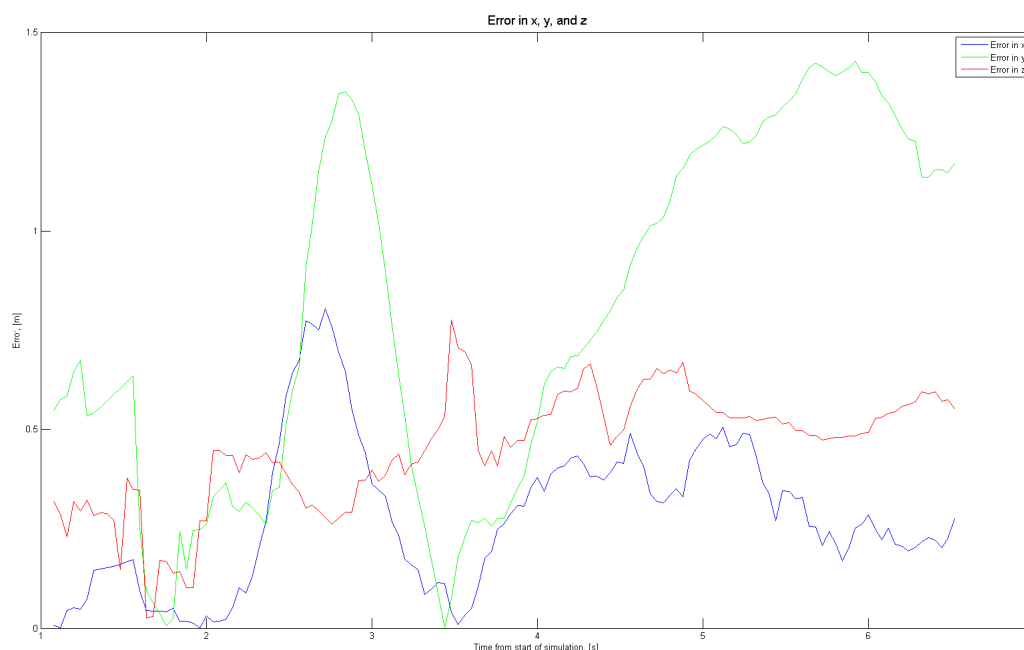
Figur 31. Två fordon färdas längs en väg i Norrköping. I den första bilden har den gröna bilen inte detekterats. Osäkerheten för det predikerade målet med identitetsnummer 79 är därför stor vilket illustreras av cirklarna som omger prediktionen. I den andra bilden har den gröna bilen detekterats och osäkerheten gällande dess tillstånd har därmed minskats.

Felet mellan den verkliga banan för ett fordon och det predikterade målsåret varierar under simuleringen. Figur 32 visar den predikterade banan och den verkliga banan för ett fordon.



Figur 32. Predikterad (röd) och verklig (grön) bana för ett fordon under en del av simuleringen i Norrköping. När fordonet svänger blir felet mellan den predikterade och den verkliga positionen något större.

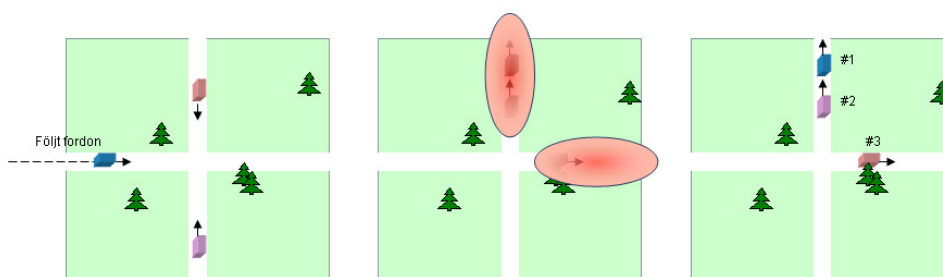
I figur 33 visas skillnaden i x-, y- och z-koordinaterna mellan den predikterade positionen och den verkliga positionen. Mellan 2 och 4 sekunder efter det att simuleringen startat svänger fordonet. Felet i x- och y-led blir då något större än innan fordonet svänger. Maximalt är felet under svängen ungefär 0.8 meter i x-led och 1.4 meter i y-led. Efter svängen fortsätter felen i x- och y-led att vara relativt stora. Detta beror på att detektionerna efter svängen inte är centrerade på fordonet, vilket de varit innan svängen. Felen efter svängen är därför något missvisande.



Figur 33. Skillnaden i x-, y- och z-koordinater mellan den predikterade positionen och den verkliga positionen. Mellan 2 och 4 sekunder efter simuleringens start svänger fordonet, vilket orsakar en ökning av felen i x- och y-led. Efter svängen är inte detektionerna, som ligger till grund för målföljningen, centrerade på fordonet vilket orsakar ett konsekvent fel i x- och y-led.

5.4 Återigenkänning

Syftet med funktionen för återigenkänning är att identifiera nyligen upptäckta fordon. När multimålföljaren hittar ett nytt mål och tilldelar detta ett id-nummer, ska alltså återigenkännaren undersöka målet och avgöra om det är ett fordon som observerats tidigare och i så fall vilket. Detta är nödvändigt för att upprätthålla korrekta fordonsidentiteter i situationer där sensorn inte kontinuerligt kan följa ett visst fordon, till exempel för att något annat område måste avspansas, eller på grund av skylande objekt eller terräng. Ett exempel på ett sådant scenario visas i figur 34, där sensorn tittar bort just när det följda fordonet passerar korsningen (bilden i mitten).



Figur 34. Ett scenario där återigenkänning är nödvändigt för att följningen ska kunna återupptas.

Återigenkänning kan baseras på flera olika typer av attribut, som till exempel utseende, rörelsemönster, storlek eller andra geometriska särdrag. Metoden som har studerats här identifierar fordon genom att undersöka deras utseende.

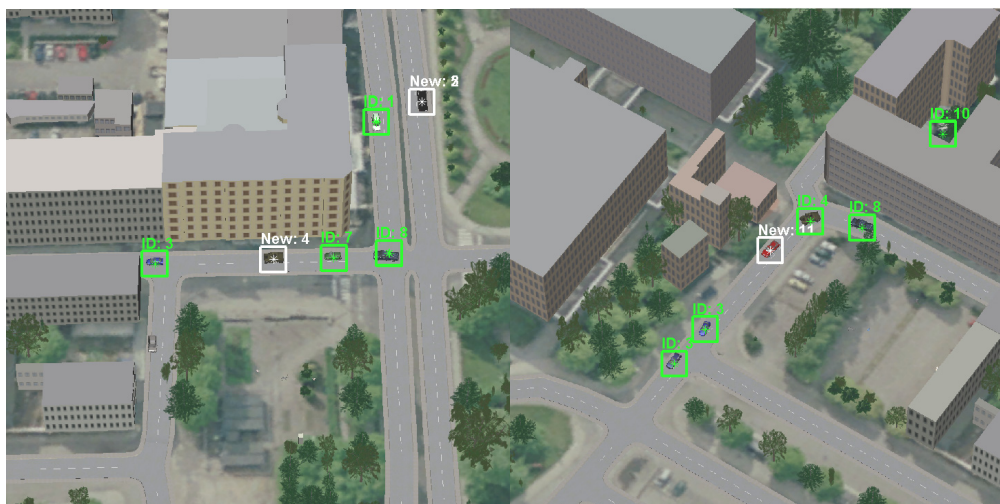
Det är viktigt att notera att syftet med återigenkänning är att upptäcka om ett fordon är identiskt med något tidigare observerat mål, och att databasen med möjliga fordonstyper alltså byggs upp efter hand. Detta skiljer sig från generell fordonsidentifiering, där varje observerat fordon matchas mot en på förhand känd signaturdatabas för att avgöra dess typ. Det är också viktigt att notera att återigenkännaren i allmänhet inte kan se skillnad på olika individer av samma fordonstyp.

Återigenkännaren har en stark koppling till multimålföljaren. Denna skickar kontinuerligt ut information om samtliga mål som följs. Varje mål identifieras i dessa meddelanden av ett unikt id-nummer som tilldelats av målföljaren. För varje mål anges också dess senaste position i världskoordinater (var i världen fordonet antas befinna sig) och i sensorkoordinater (var i kamerabilden målet har observerats). Återigenkännaren lyssnar på informationen från målföljaren, och så snart ett nytt id-nummer tilldelats, undersöks den del av bilden där det nya fordonet finns. Ett antal olika särdrag som beskriver fordonet extraheras och jämförs med samtliga tidigare observerade fordon. Slutligen skickar återigenkännaren ut ett meddelande om fordonets identitet. När ett nytt fordon observerats, sparas dess särdrag i återigenkännarens interna databas för senare återigenkänning och återigenkännaren sänder ut ett meddelande om att en ny fordonstyp hittats.

Om återigenkännaren inte kan identifiera fordonet, till exempel på grund av att det ses från en ovanlig vinkel, skickas i stället ett meddelande om detta till sensorstyrningen. Den kan då välja att styra sensorn så att återigenkänningen underlättas, om detta vid tillfället är en prioriterad uppgift.

Återigenkännaren består i huvudsak av två delar. Dels en *ensning* som utgår från en bild av ett fordon och från målföljarens information om fordonet, och normaliserar bilden så att alla fordon avbildas i samma skala och på samma position. Dels en *klassificering* som extraherar ett antal särdrag som beskriver fordonet och jämför dessa med en databas med andra observerade fordon. Klassificeringen beskrivs i detalj i [19].

För det studerade scenariot har återigenkänningen fungerat ganska bra. Figur 35 visar två bilder från Norrköpingsscenariot, strax innan och efter det att en fordonskolonn skyls bakom ett hus. Medan fordonen inte syns ansluter en ny bil (nummer 11) till kolonnen medan en av de ursprungliga (nummer 7) parkerar bakom huset. Detta detekteras korrekt av återigenkännaren. Samtidigt identifieras en stillastående bil (långt ner till vänster i den andra figuren) felaktigt som nummer 3. Detta eftersom den liknar den ursprungliga bil 3 till både färg och form.



Figur 35. Återigenkänning av fordon i Norrköping strax innan och efter det att en kolonn skyls bakom en byggnad. Fordonet med id-nummer 11 har tillkommit till kolonnen medan nummer 7 parkerat bakom byggnaden.

Resultatet från återigenkännaren med den visuella sensorn är 92 % korrekt klassificerade fordon av totalt 18 typer. I algoritmen görs ingen ensning för att kompensera för variationer i fordonens riktning. Klassificeringen fungerar bra ändå eftersom fordonen i stor utsträckning har olika färg. Om en operatör endast är intresserad av att leta efter fordon som liknar en van och/eller pickup kan den välja att bara visa ID:2 och/eller ID:8. Detta kan hjälpa en operatör att snabbare sälla fram de misstänkta fordonen bland alla fordon i scenariot. Detta kan även underlätta vid följning av ett eller flera fordon.

På grund av begränsningar i informationen från målföljaren kan vi i nuläget inte normalisera bilderna med avseende på fordonens orientering, vilket försämrar igenkänningsprestanda. När vi har tillgång till information om fordonens färg, fungerar igenkänningen bra trots detta. Ensning av orientering kommer att implementeras inom kort.

5.5 Ruttplanering och sensorstyrning

Att följa flera fordon samtidigt i en stad, utan att tappa bort dem, är ett svårt problem. UAV-systemets flygbana och styrning av sensorn måste då planeras så att ocklusion från byggnaderna minimeras samtidigt som målspåren för de intressanta målen uppdateras tillräckligt ofta så att de inte tappas bort. För att lösa detta har ruttplanering och sensorstyrning använts. I denna studie består planeraren av två delar, den första delen planerar flygrutten givet att UAV-systemet är utrustad med en sensor med obegränsat synfält. Nästa del utgår ifrån denna flygrutt och planerar i vilka riktningar sensorn ska peka.

I planeringen kan inte 3D modellen användas för att beskriva ocklusionen; d.v.s. från vilka riktningar vägarna kan ses och inte. Istället används en enklare representation som kan beskrivas som "murar" med olika höjd längs med alla vägar. Höjden på muren är den höjd som byggnaden bredvid vägen har, finns ingen byggnad nära vägen så är höjden noll. Denna representationsform är en grov förenkling, men kan göras mycket snabb att räkna på vilket är ett krav för planeringsalgoritmen.

Planeringshorisonten för detta scenario har varit 3 sekunder vid varje planeringstillfälle och båda planeringsdelarna försöker minimera positionsosäkerheten för alla fordonen givet att varje mål är representerat med ett "partikelmoln". En "partikel" kan ses som en hypotes om var fordonet befinner sig och vilken hastighet den har. En gradientsökningsmetod används för att lösa det optimeringsproblem som planeringsproblemet resulterar i. En nackdel med en sådan metod är att den kan få problem när målen är alltför separerade, och inte minst när ocklusion från byggnader och vegetation är påtaglig. Ett knep för att inte planeringen enbart ska fokusera på bara en grupp av mål, är att välja startgissningen i gradientsökningen på ett smart sätt. I det aktuella fallet väljs startgissningen så att sensorn riktas mot den förväntade positionen för alla fordonsgupper i tur och ordning.

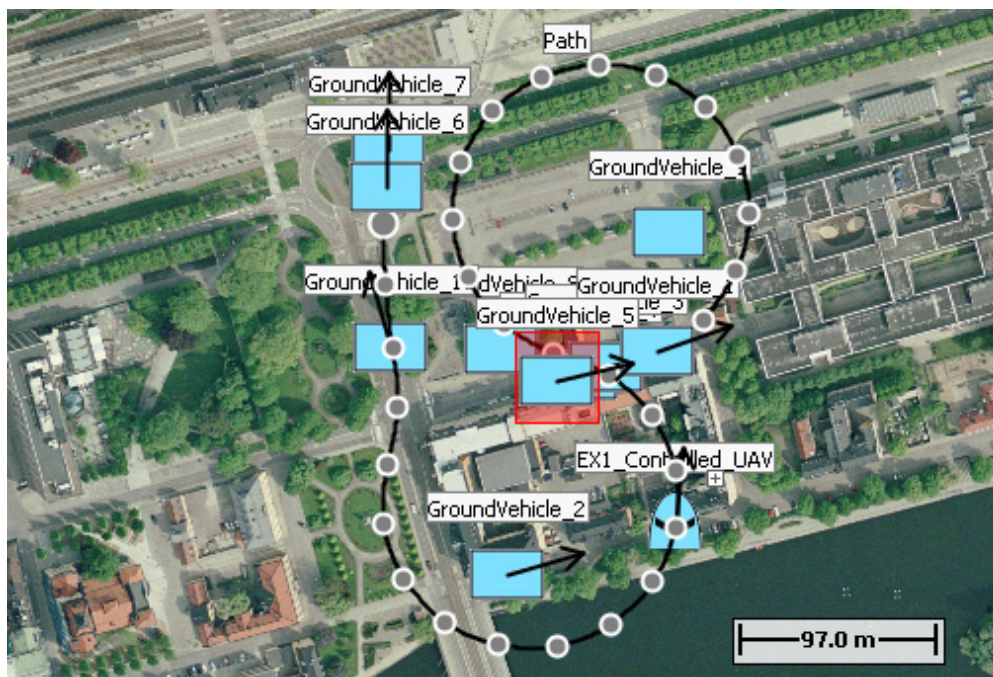
UAV-modellen som har använts vid planeringen är en enkel så kallad "coordinated turn model", med konstant hastighet (här 30 m/s) och flyghöjd (här ca 600 m över marken). Styrsignalen är UAV-systemets förändring i orientering (heading). Modellen för sensorgimbalen består av två enkla första ordningen modeller för pan respektive tilt, där styrsignalen är vinkelförändringen för respektive axel. En detaljerad beskrivning av planeringsalgoritmer för UAV-system med styrbar EO/IR-sensor ges i [20].

Figur 36 visar hur de misstänkta fordonen tar olika vägar: Pick-up:en svänger till vänster medan den svarta GMC Van åker rakt fram och svänger vänster i nästa korsning.



Figur 36. De misstänkta fordonen tar olika vägar. Pick-up:en svänger till vänster medan den svarta GMC Van åker rakt fram och svänger vänster i nästa korsning.

Flygbana och styrning av sensorn har beräknats för detta scenario. I figur 37 visas hur UAV-systemet ska flyga för att göra det möjligt att följa både pick-up:en och van:en som befinner sig på olika sidor om ett kvarter. Både UAV- och gimbalmodellerna har styrsignalsbegränsningar och UAV-systemet ligger oftast på minimal svängradie.



Figur 37. Planerad rutt för att möjliggöra följning av både pick-up:en och van:en som befinner sig på olika sidor om ett kvarter.

I figur 38 visas de båda misstänkta fordonen. Van:en är i figur 37 "GroundVehicle_2" och ses från UAV:ns position. Pick-up:en är "GroundVehicle_5" men syns inte från UAV-systemets position.



Figur 38. De misstänkta fordonen på varsin sida om ett kvarter.

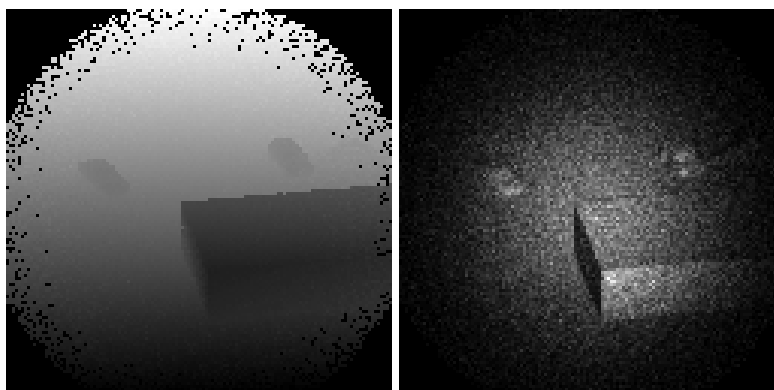
5.6 3D-laser

Uppgiften för det andra UAV-systemet är att samla in information om två fordon, en Toyota RAW4 och en BMW E36. Dessa står parkerade utanför en byggnad i centrala Norrköping, vilket visas i figur 39.



Figur 39. Toyota Raw4 till höger och BMW E36 till vänster.

Figur 40 visar resultatet av lasersimuleringen med de två fordonen parkerade bakom byggnaden. I detta fall har simuleringsparametrar valts enligt beskrivningen i kapitel 4.1 för att efterlikna den amerikanska sensorn ASC Flash 3D. Den stora styrkan med denna lasersensor är att den enbart behöver ett laserskott för att göra en hel 3D-mätning av scenen. Andra lasersystem är exempelvis *skannande* och mäter in scenen successivt genom att skott för skott bygga upp en hel 3D-bild. Den amerikanska ASC Flash 3D sensorn utnyttjar en array om 128 x 128 detektorer för att momentant mäta upp scenen. Detta kan den göra med en maximal uppdatering på 30 bilder per sekund. På så sätt uppstår inga problem vid snabba händelseförlopp där rörelseoskärpa annars förkommer med skannande system som kräver längre tid för varje avbildning.



Figur 40. Avståndsbild och reflektansbild från lasersimuleringen.

Lasersystemets fördelar gentemot en 2D-sensor är att det är möjligt att estimerar fordonens dimension och att noggrant mäta deras position direkt från inmätt avståndsdata. Eftersom sensorn är aktiv, dvs. den sänder ut och tar emot sin egen strålning, blir den oberoende av dagsljus och mindre känslig för väder.

6 Diskussion

6.1 Nyttiggörande

Simuleringsmiljön som vidareutvecklas har under 2009 använts av flera projekt för sensorsimuleringar. Några exempel på projekt är Autonom spaning, ARCUS, Autonoma funktioner och Taktikanpassning SAT, vilka alla har bidragit på olika sätt till MSSLab.

Under året har projektet bidragit med underlag till ett antal Försvarsmaktsstudier. Projektet har delvis stöttat studierna ekonomiskt inom ramen för vad FoT-projekt är ålagda att göra, men framförallt genom att bidra med den kompetens som byggts upp inom detta och andra pågående projekt. Tekniker och kompetenser som utnyttjats omfattar sensorikunskap och därmed tillhörande signalbehandling inom IR, laser och radarområdet.

6.2 Fortsatt arbete

MSSLab har redan idag mycket bra funktionalitet, men mycket arbete återstår för att en generell simuleringsmiljö skall erhållas. Den snabba teknikutvecklingen inom simuleringsområdet möjliggör ny funktionalitet och förbättrad sensorsignalkvalité.

Tillgängliga omvärldsmodeller har god kvalité, men de bör förbättras och kompletteras. Idag är det bara Kvarnmodellen som har olika texturer såsom IR, visuell och materialklassad textur. Norrköpingsmodellen har endast visuell textur. Under 2009 genomfördes en mätkampanj med ett flygplan som var utrustat med både laser och hyperspektral sensor. Dessa data kommer under nästa år att användas för att materialklassificera Norrköpingsmodellen. Detta för att göra det möjligt att simulera radar och IR-system i en urban miljö.

Arbetet med animering av människor bör fortsätta för att göra det möjligt att simulera stora grupper av människor. Fler rörelsemönster bör även mätas upp för att göra det möjligt att simulera människor som placerar ut väskor, slåss, ramlar etc.

IR-simuleringarna som gjordes 2008 i olika väderfall behöver valideras ytterligare. En metod för detta är att jämföra de parametriska atmosfärerna som räknats fram i CameoSim via MODTRAN med körningar direkt i MODTRAN, samt jämföra med teoretiska beräkningar direkt från storheternas definitioner. Kontraster i CameoSim-bilder kan även jämföras med mätningar.

En begränsning med MODTRAN är att molnen i modellen täcker hela himlen. Det går alltså inte att ändra molnens täckningsgrad på himlen. Ett sätt att lösa detta kan därför 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 vattenintensitet.

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

Norrköpingsmodellen är en mycket bra urban terrängmodell, men den behöver förbättras. Idag har alla hus i omvärldsmodellen platt tak, men genom att kombinera laser data och shape data bör taktyp kunna bestämmas såsom snedtak,nocktak etc. I shape data finns info om byggnadstyp såsom industrilokal, bostadshus etc. De visuella texturerna kan förbättras genom att några olika visuella texturer tas fram för olika byggnadstyper såsom industrilokaler, bostadshus etc. Att estimerar formen på byggnaderna genom att använda shape data och laserdata fungerar inte så bra på vissa byggnader som t.ex. kyrkor. Här kan modelleringsjobb vara nödvändigt.

7 Referenser

- [1] Internet: <http://www.triangraphics.de>, besökt 2009-12-01.
- [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] Näsström F., Karlsson M., Carlsson L., Chevalier T., Follo P., Hedström J., Karlsson N., Persson A., Sume A., "Simuleringsbaserade metoder för sensorvärdering", FOI-R--2629--SE, Linköping, December 2008.
- [5] Felker, F and Light, J, "Aerodynamic Interaction between a Rotor and Wing in Hover", 42nd AHS Annual Forum, Washington D. C., June 1986.
- [6] Internet: <http://home.gna.org/cal3d>, besökt 2009-12-01.
- [7] Internet: <http://www.opengl.org>, besökt 2009-12-01.
- [8] Internet: <http://www.openscenegraph.org/projects/osg>, besökt 2008-12-09.
- [9] Larson H., "Test plan for ASC FLASH 3D Ladar trials in Sweden, September 2008", FOI-D--0310--SE, Linköping, September 2008.
- [10] Chevalier T., "Statusrapport Modellering av 3D-avbildande laser", FOI MEMO 2417, Linköping, Maj 2008.
- [11] Chevalier T., "Modeling of 3D laser radar systems. Status of FOI-LadarSim in May 2009", FOI-R--2764--SE, Linköping, Maj 2009.
- [12] Bennet F. och Fenelius S., "SceneServer - a 3D software assisting developers of computer vision algorithms", FOI-R--0831--SE, Linköping, Mars 2003.
- [13] Internet: <http://www.openscenegraph.org>, besökt 2007-12-16.
- [14] Michael W., "Light Space Perspective Shadow Maps", Vienna University.
- [15] Gundmark T., Dalberg E., Svenson P., Lindgren D., Grahn P., Linder S., Hansson A., Kylesten B., "Ledningssystem – demonstration och värdering av teknik och metodik, Slutrapport", FOI-R--2645--SE, Linköping, December 2008.
- [16] Karlholm J., "Implementation and evaluation of a method for detection of ground targets in aerial EO/IR imagery", FOI-R--1267--SE, Linköping, June 2004.
- [17] Viola, P. and Jones, M. J. (2001). Robust real-time object detection. Technical Report CRL 2001/01, Cambridge Research Laboratory, Compaq Computer Corporation.

-
- [18] Näsström F., Cirkic M., Rydell, J., Skoglar P., Stenborg, K.-G., Ulvklo, M., "Autonoma funktioner för intelligenta operatörsstöd". FOI-R--2752--SE, Linköping, Mars 2009.
- [19] 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, Linköping, Oktober 2008.
- [20] Skoglar, P. Planning Methods for Aerial Exploration and Ground Target Tracking, Linköping Studies in Science and Technology. Thesis 1420. 2009.