Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

AN APPROACH TO DESIGN AND DEVELOPMENT OF DECENTRALIZED DATA FUSION SIMULATOR

Chandresh Mehta
Govindarajan Srimathveeravalli
Thenkurussi Kesavadas

Virtual Reality Lab
809 Furnas Hall
SUNY at Buffalo
Buffalo, NY 14260, U.S.A.

ABSTRACT

This paper discusses the ongoing efforts on development of a *Decentralized Data Fusion (DDF)* simulator for analysis and design of a distributed fusion-based tracking system. We have identified the requirements for a DDF simulator and have developed a fully interactive, graphical user interface based scenario generation tool called *SceneGen* (Srimathveeravalli, Subramanian and Kesavadas 2004) for creating battlefield scenarios, and a simulation tool called *VizSim* for running various DDF algorithms on scenarios created in *SceneGen* and displaying the simulation results in an easy to understand fashion. *SceneGen* and *VizSim* has been designed with a full compliment of user utilities, including an efficient terrain database generation module, a sensor report generation module and the database connectivity to store and retrieve scenarios and simulation results. The innovative visualization techniques used in the simulator helps in displaying the data in a fashion that transfers maximum information to a user. Both *SceneGen* and *VizSim* have been tested successfully for running scenarios consisting of a large number of entities.

1 INTRODUCTION

The DDF research at University at Buffalo is aimed at developing an integrated, simulation-based toolkit for the design and analysis of agile, decentralized fusion-based tracking systems. Such a simulator is intended to be used by Network Centric Warfare (NCW) applications for developing a Common Operating Picture (COP) for shared battlespace awareness and decision-making by friendly commanders. The architecting of a DDF simulator is a very complex process requiring the integration of target tracking, networking, data fusion and visualization. The details about the algorithms and methods used for target tracking, networking and data fusion is out of the scope of this paper. This paper reports on the requirements of a DDF simulator and the approaches taken to fulfill those requirements. It also gives specific details about the visualization techniques and the terrain database generation module of the simulator.

2 SCENARIO DESCRIPTION FOR DDF SIMULATION

A typical battlefield scenario for DDF simulation consists of air targets and ground targets, which are observed by sensors on air platforms and ground platforms. Platforms in scenario are partitioned into groups called clusters and within each cluster a platform (clusterhead) is chosen, using a network topology algorithm, to receive information from other platforms in the cluster. Each sensor runs a tracking algorithm to generate tracks for the observed targets and communicates them to its clusterhead. The clusterheads use fusion algorithms to fuse the tracks from various sensors to provide the user with COP.

3 REQUIREMENTS OF A DDF SIMULATOR

We have identified following requirements for a DDF simulator.

3.1 Scenario Generation

Utilizing real life battle scenarios as data sources for DDF simulation is unsuitable, as data acquired is not ideal and would require refinement to suit development and testing of decentralized fusion concepts. At a design and development stage there is a need for a scenario generation system that facilitates generation of test scenarios, for use in the simulation testbed. The creation of scenario should be easy, interactive and intuitive. The user should have the option of creating a scenario by an automated or manual process. In the automated case a batch of scenarios will be generated by simply selecting certain parameters, while in the later case the user would create the scenario one sensor or target at a time through the user interface. Prior to the creation of the scenario the user should be able to define simulation variables like simulation time of the scenario and the time step interval.

3.2 Terrain Database Generation

A DDF simulation is closely tied with geospatial information. Terrain features and cultural artifacts such as roads, vegetation etc. influence the behavior of ground targets. Hence it becomes imperative that a database which includes this information be integrated into the testbed to permit realistic simulation. There should be provision for importing realistic Geographical Information System (GIS) and terrain data in testbed for any true world location for battlefield referencing.

3.3 Sensor Models

Sensors contribute to battlespace awareness and can be a human or a device deployed on air or ground based platforms. The testbed should be capable of modeling behavior of various fixed sensors such as Unattended Ground Sensor (UGS) as well as moving sensors such as radar and Ground Moving Target Indicator (GMTI). In addition to this there should be means for specifying the attributes of these sensors while creating the scenario. A sensor report generation module should be capable of line of sight calculations to check if a target is obscured from a sensor during the span of simulation due to terrain or GIS entities like buildings and foliage.

3.4 Target Models

The enemy targets can be air based such as an aircraft or ground based such as battle tanks, radar guns, surface to air missile launch vehicles and human. The testbed should allow selection of these target models for scenario creation and simulation. In addition to this there should be means for defining a complex motion behavior for a target as well as specifying its attributes.

3.5 Platform Models

Platforms are sensor carriers and can be an aircraft such as AWCAS, UAV, Predator, Global Hawk etc. or human. The testbed should have a provision to define a platform in scenario as a probable clusterhead and specify its communication parameters. The user should be able to define a complex maneuvering path for platforms by selecting few waypoints.

3.6 Database Support

There is a great monetary and time cost involved in the development of scenarios; hence a provision for the reuse of scenarios. This means that the scenario generated should be saved and made available for later reuse and edits. In addition to this, simulation results should be saved for performance evaluation. The size of scenario should not be a limiting factor during the creation of scenarios. In most situations, a database support must be provided for effective storage and maintenance of the scenario data. The database support should allow storage of entity behavior and model information and storage of data generated during the simulation itself.

3.7 Simulation and Information Visualization

The testbed should be capable of displaying simulation results at a rate that is easy for the user to absorb and should be able to give a visual representation of platform specific information such as model type, number of sensors deployed, tracking algorithm used etc. so that friendly commanders can use it as a decision-making tool.

4 PRIOR WORK

Department of Data and Information Fusion of Swedish Defense Research Agency have developed a system called Infofusion demonstrator 03 (IFD 03) (HÖrling, Mojtahed, Svensson and Spearing 2002; Svensson and HÖrling 2003) for information fusion processes. However their demonstrator aims only at simple ground warfare scenarios. IFD uses commercial scenario generator from FLAMES™ (FLAMES 2004). This scenario generator supports incorporation of sensor and target modules through the usage of scripts (Nicholson, Lloyd, Julier and Uhlmann 2002; Svensson and HÖrling 2003). It incorporates a terrain database generator and also supports consideration of terrain masking characteristics in the scenario.

The Lockheed Martin Advanced Technology Laboratories (ATL) has developed a real-time, onboard/offboard multi-sensor, multi-target data fusion system (Pawlowski and Gerken 1998). They have created a graphical Data Fusion Input Simulator (DFIS) to generate scenarios and a Data Fusion Workstation (DFWS) to display simulation results. This system lacks support for GIS data.

Our collaborating group (Kim, Singh and Llinas 2002) had previously developed a proof of concept simulator for distributed target tracking. A simple scenario generation module using Matlab was developed to support this simulator. However, this scenario generator was found to be insufficient for generation of realistic scenarios as it lacked terrain support, and could not satisfy the previously listed requirements of scenario generation.

Research groups in United States military and the Air Force have also addressed the need for a scenario generator to support the development of their DDF testbeds. Development work for this need is reported in Blais, Brutzman, Horner and Nicklaus (2001) and Dawidowicz, Tran, Wong, Mayk and Klose (1997). These groups have addressed some of the requirements that we have enlisted in the previous section. However both the groups have reaffirmed the need for terrain support and 3D visualization in their work.

Other commercial products are available from Boeing (Battlescape 2004) and TNO-FEL (HAREM 2004) for the purpose of battlefield simulation and scenario creation. However these applications seem to be very special purpose in nature and little information is available about the architecture, utilities, modules etc.

5 TESTBED ARCHITECTURE AND FUNCTIONALITY

5.1 Software Architecture

Keeping in mind the requirements listed in section 3, we have created a comprehensive DDF testbed by developing two softwares viz. *SceneGen* and *VizSim*. *SceneGen* is a scenario generation tool and is used for creating scenarios for use in DDF testbed. *VizSim* is a simulation tool and is used for running DDF algorithms and displaying simulation results in an easy to understand fashion.

Our application has been developed using C++ in a Linux environment. Open Inventor API (Coin 2004) is used for graphics rendering. The user interface has been developed using Trolltech's Qt (Qt 2004) GUI development toolkit. MySQL is used for data warehousing and the KDevelop IDE is used as the development environment.

The following sections describe some of the design issues and approaches taken in the development of *SceneGen* and *VizSim*.

5.2 SceneGen: Scenario Generation Tool

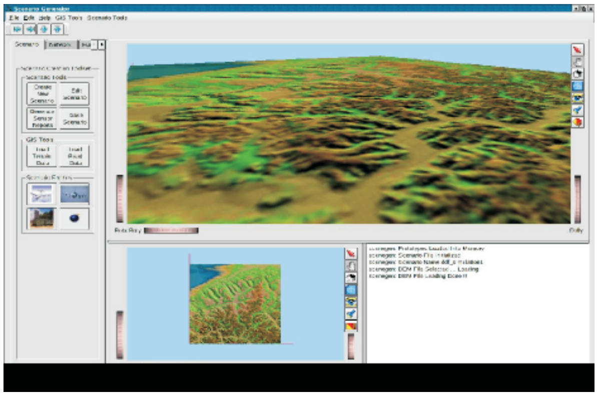
SceneGen has an easy to use Graphical User Interface (GUI) which allows interactive creation of a scenario and generation of ground truth. The user has the capability to edit scenario entities while creating a scenario; this adds additional flexibility to the scenario creation process. The ground truth obtained from the scenario generator is used to 'seed' the DDF algorithms. The following sections describe various functionalities of *SceneGen*.

5.2.1 Terrain Database Generation

We have developed an efficient terrain database generation module for importing true terrain and GIS data into the testbed.

<u>*A. Terrain Data*</u>

SceneGen takes Digital Elevation Model (DEM) files as input for getting terrain data of the desired location. Apart from being commonly used another advantage associated with using DEM files is that it can be freely downloaded from USGS repositories. Rendering is very expensive if one uses raw data from the DEM files. To tackle this issue, we triangulate the data to compute TIN that gives the terrain skin as shown in Figure 1. The user inputs error threshold for the terrain and maximum number of points to be included in triangulation. The greedy insertion algorithm (Garland and Heckbert 1995) that we used for triangulation minimizes both the error and the number of triangles in the mesh and efficiently renders a terrain.



```
scenegen: Prototypes Loaded Into Memory
scenegen: Scenario File Initialized
scenegen: Scenario Name ddf_s Initialized
scenegen: DEM File Selected ... Loading
scenegen: DEM File Loading Done!!
```

Figure 1: Triangulated Irregular Network

<u>*B. Vector Data*</u>

The 2D vector maps providing cultural and other important data such as roads, rivers, foliage can be imported in the testbed from USGS DLG files which are freely available from USGS repository. One of the issues with associating terrain and vector data is disparate coordinate system. Whereas the terrain data is in geographical coordinate system, the vector data is available in UTM coordinate system. Our scenario generator smoothly takes care of this disparity by accurately projecting vector data in the geographic coordinate system as seen in Figure 2.

Another issue in terrain database generation is overlay of 2D vector data over 3D terrain data. The 2D vector data cannot be simply draped onto the 3D terrain as it will cause z-buffer artifacts (Kersting and DÖllner 2002). We have efficiently overcome this problem. For accurate draping we test the elevation at each node point. However for long straight sections of road if we check only at the node point, there is still a chance of road-ground intersection if the terrain is not flat in that area. To avoid this we test the ground along entire road segment and subdivide the road as needed.

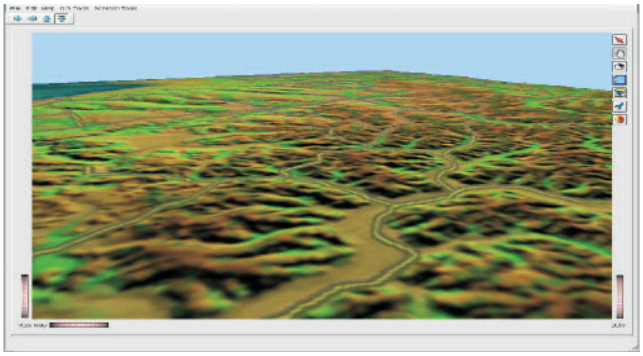


Figure 2: GIS Data, Initially in UTM Coordinate System, is Projected in Geographical Coordinate System And is Accurately Draped on the Terrain

5.2.2 Sensor Modeling and Line of Sight Calculations

SceneGen supports modeling of radar sensors for air target tracking and GMTI sensors for ground target tracking. Figure 3 shows a radar sensor being deployed on an air platform. The red sphere shows the coverage range of the sensor.

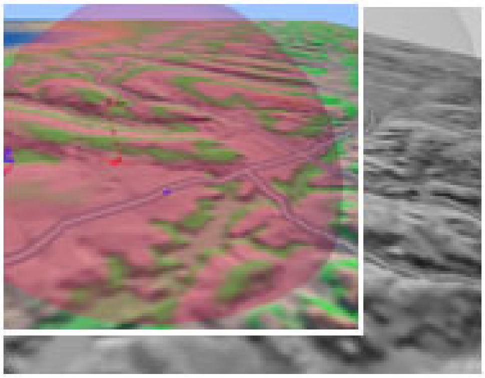


Figure 3: Radar Sensor Observing Air Target

Figure 4 shows a GMTI sensor being deployed on a platform in scenario to do surveillance on ground based targets. A sensor report generation module generates reports for these sensors after testing for obscuration of target due to terrain and GIS entities. We employ basic ray tracing methods for this test. Currently we are working on modeling SIGINT sensors.

5.2.3 Target and Platform Modeling

SceneGen supports the creation and modeling of a number of air-based and ground-based targets and platforms. The prototype information of various kinds of aircraft, tanks, missile launch vehicles and radar-guns are provided in the database. The user can select the prototype information from a database and use that prototype information to model the entity that needs to be used in scenario. The user is provided with a form (Figure 5) from which the entity prototype can be chosen and additional platform specific information can be entered. Any given platform can support a number of communication and sensor units on it. A vehicle dynamics engine enables a complex path generation for air and ground entities. The ground entities can be constrained to move on-road.

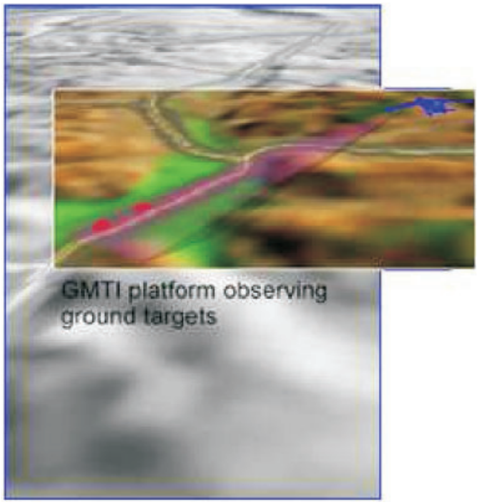
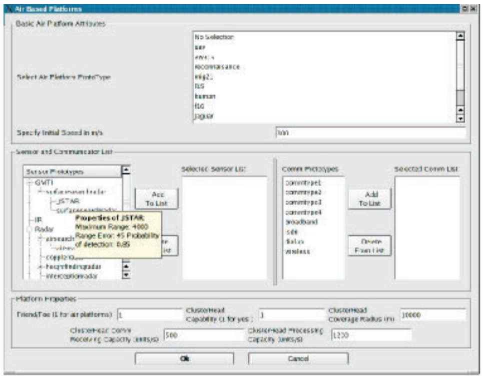


Figure 4: GMTI Sensor Observing Ground Targets



Air Based Platforms

Basic Air Platform Attributes

Select Air Platform ProtoType: No Selection, aar, awacs, reconnaisance, mig21, tus, human, tu6, jaguar

Specify Initial Speed in m/s: 100

Sensor and Communicator List

Sensor Prototypes: GMTI, surfacesearchradar, JSTAR, surfacesearchradar, IR, Radar, airsearch, airsearch, copplerradar, heightfindingradar, interceptionradar

Properties of JSTAR:
Maximum Range: 4000
Range Error: 45 Probability of detection: 0.85

Selected Sensor List

Comm Prototypes: commtype1, commtype2, commtype3, commtype4, broadband, sde, status, wireless

Selected Comm List

Platform Properties

Friend/Foe (1 for air platforms): 1

ClusterHead Capability (1 for yes): 1

ClusterHead Coverage Radius (m): 10000

ClusterHead Comm Receiving Capacity (units/s): 500

ClusterHead Processing Capacity (units/s): 1200

Figure 5: Form for Selecting Platform Properties

5.2.4 Database Support and Reusability

The scenario generator uses a MySQL database for data warehousing. The database is used for storage of data generated during the scenario and for storage of entity model and prototype information library. The use of the database allows for easy storage and retrieval of scenarios. The database facilitates loading and editing of previously created scenarios. The database schema can be changed to store data in different formats for different information fusion testbed needs.

5.2.5 Automatic Scenario Creation

Scenario can be built in automatic mode in which user sets certain parameters required to built a scenario and then all the steps, i.e., initializing the database, selecting platform and target properties, picking ground truth of platforms and targets, generating sensor reports and writing scenario to database is done by Scene Generator without any need for user intervention. This mode can be used to create a batch of scenarios, i.e., more than one more scenario with identical platforms but different ground truth, which is useful in Monte-Carlo analysis.

5.2.6 Information Visualization

It is necessary to represent the entities and data in fashion that transfers maximum information to a user at a glance, but at the same time can also lend itself for in-depth analysis. For this purpose we have used three 3D icons and symbols to represent information and have provided mouse rollover feature for displaying entity specific information such as prototype, number of sensors and communicators deployed on platform etc as shown in Figure 6.

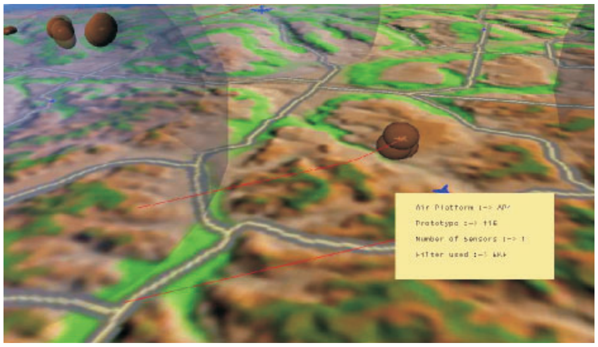


Figure 6: Information Visualization

We have followed traditional conventions for the representation of targets (red color) and friendlies (blue), but at the same time have adopted further concepts for iconic representation. We have used transparency, shape, size and shininess of various icons as metrics for information transfer. For example, consider the representation of the sensor range (Figure 3), the sensor range is represented by a simple spherical shape in *SceneGen*, the information about the sensing range of the sensor is given by its size and the sensing strength of the sensor is represented by the transparency (least transparent is most powerful and the most transparent is the weakest sensor). Entity and target trajectory is another important data that needs to be conveyed effectively to the user. This information is represented as three dimensional curves and lines on a temporal basis. We use high resolution VRML models of various aircraft and ground sensors to represent the entities. New models can be easily included and old models edited if desired.

5.3 VizSim: DDF Simulation Tool

The *VizSim* interface allows the user to monitor the progress of a DDF simulation at runtime. The scenario generated in *SceneGen* forms the input to *VizSim*. The DDF algorithms of networking, tracking and fusion are implemented in this software. It incorporates all information visualization and terrain database support developed for *SceneGen*. The following sections describe additional functionalities of *VizSim*.

5.3.1 Simulation Parameter Settings

The user can modify critical parameters relevant to tracking and networking algorithms before executing the simulation. The user has the capability to select from multiple tracking and networking algorithms for simulation. A sensor specific design approach used in *VizSim* enables selection of a different tracking algorithm for each sensor.

5.3.2 Visualization of Simulation Results

VizSim displays tracking, networking, and fusion results in an easy to understand fashion.

The user has the capability to view the simulation results from any desired viewpoint and then manipulate the environment to adaptively extract and examine regions of interest thereby enabling focused decision-making.

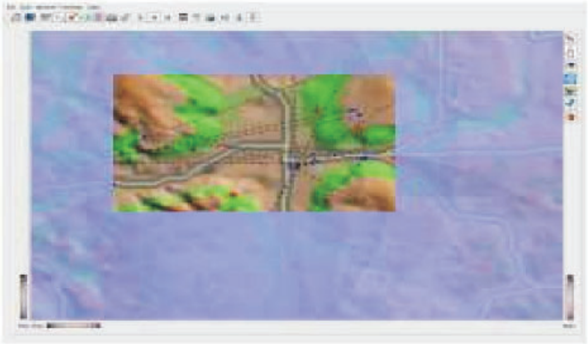


Figure 7: Network Topology Results

Figure 7 shows results of network topology module. The black dashed line indicates the communication linkage between a platform and its clusterhead.

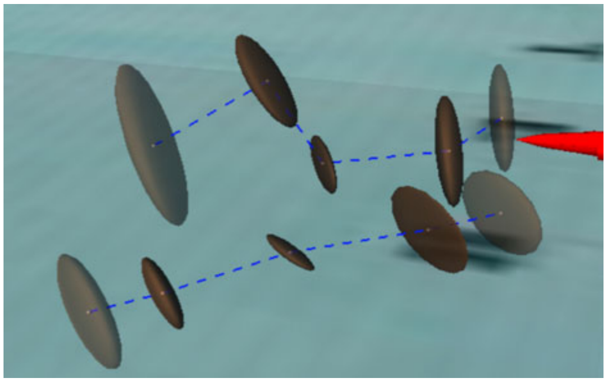


Figure 8: Visualization of Tracking Results

Figure 8 shows the screenshot of tracking results being visualized in simulation software. The ellipsoids are the sensor covariance and the dashed lines are the sensor tracks.

5.3.3 Database Support

Similar to *SceneGen*, *VizSim* also uses a MySQL database system for data warehousing. The simulation results are stored in database for the purpose of finding performance metrics of DDF algorithms.

6 RESULTS

The application was implemented on a Pentium 4 desktop with 512 Mb RAM, a 64 Mb NVIDIA GeForce graphics card with Red Hat Linux 9 operating system. The testbed was then loaded up with a 1 degree DEM of Buffalo City, New York. Road data of entire region was draped on to the terrain from 16 different DLG files. The roads were found to be properly associated with the terrain. A test scenario was then created with 7 ground based platforms and 7 air based units carrying various sensor (Radar and/or GMTI) specified by the users. 7 air targets were created as two groups flying in formation with crossing trajectories. 7 ground based targets were placed in scenario with their motion constrained to on road path. Four ground platforms and four air platforms were designated with clusterhead capabilities. The sensor reports were generated and the data generated were stored in a database. Figure 9 shows the screenshot of test scenario created in *SceneGen*.

The test scenario created in *SceneGen* was opened in *VizSim* for simulation. An Extended Kalman filter tracking algorithm was selected for all the sensors in the scenario. Column Generation algorithm was selected for network topology management. The various DDF algorithms were found to perform efficiently. Figure 10 and Figure 11 shows the screenshot of simulation window.

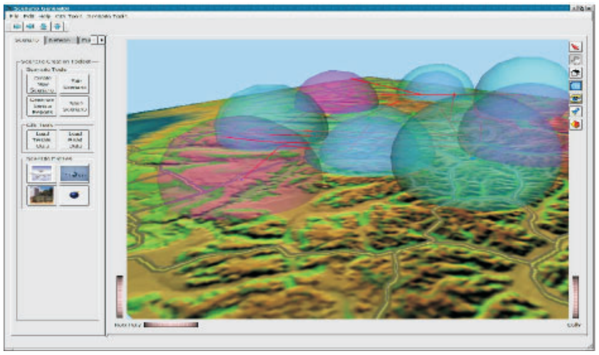


Figure 9: Test Scenario

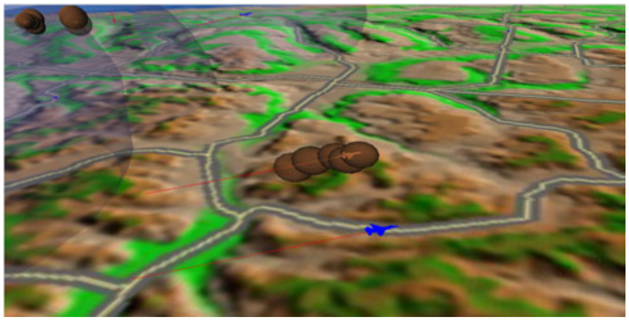


Figure 10: DDF Testbed Simulation Window Showing Air Target Tracking Results

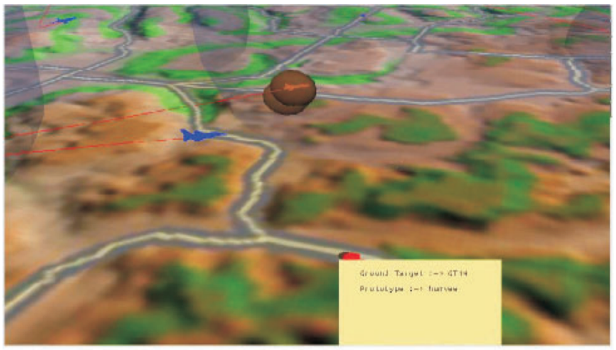


Figure 11: DDF Testbed Simulation Window Showing Ground Target Tracking Results

7 CONCLUSIONS

We have described the features and architectural details of DDF testbed used for the design and analysis of agile, decentralized fusion-based tracking systems. *SceneGen* and *VizSim* were developed to satisfy common requirements expected out of a DDF testbed. Both the software has been used extensively for evaluation of DDF system. The modular architecture of these softwares was found to be flexible and extensible to variety of adjacent tactical domains.

ACKNOWLEDGMENTS

This research was performed under the sponsorship of NIMA and managed by Rosettex Technology and Ventures Group under Task Order # RTVGC-02-216-03-C-CR-2-001. The authors gratefully acknowledge their support.

REFERENCES

BattleScape. 2004. Battle Space Visualization Software, Boeing, Available online via `<http:/washops.boeing.com>` [accessed February 2004]

Blais, C., D. Brutzman, D. Horner, and S. Nicklaus. 2001. Web-Based 3D Technology for Scenario Authoring and Visualization: The SAVAGE Project. *In Proc. Interserivce/Industry Training, Simulation and Education Conference.*

Coin. 2004. Open Inventor 3D Graphics API, Systems in Motion, Available online via `<www.coin3d.org>` [accessed August 05, 2004]

Dawidowicz, E., L. Tran, R. Wong, I. Mayk, and D. Klose. 1997. Scenario Animator and Generator as a Command and Control Decision-Making Tool for Humanitarian Assistance. *In Proc. 4th Inter-national Command and Control Research and Technology Symposium.*

FLAMES. 2004. Battlefield Simulation Software, Ternion, Available online via `<www.ternion.com>` [accessed August 05, 2004]

Garland, M., and P. Heckbert. 1995. Fast Polygonal Approximation of Terrains and Height Fields. *Tech. report CMU-CS-95-181, Computer Science Department, Carnegie Mellon University.*

HAREM. 2004. Air defense model simulator, Available online via `<http://www.tno.nl/instit/fel/div1/harem.html>` [accessed August 05,2004]

HÖrling, P., V. Mojtahed, P. Svensson, and B. Spearing. 2002. Adapting a Commercial Simulation Framework to the Needs of Information Fusion Research. *In Proc. of Fifth Int. Conf. on Information Fusion* 220–227.

Kersting, O., and J. DÖllner. 2002. Interactive 3D Visualization of Vector Data in GIS. *GIS'02*

Kim, J. J., T. Singh, and J. Llinas. 2002. Large Scale Simulation of a Distributed Target Tracking System. *In Proc. Fifth Int. Conf. Information Fusion* 624–629.

Nicholson, D., C. M. Lloyd, S. J. Julier, and J. K. Uhlmann. 2002. Scalable Distributed Data Fusion. *In Proc. Fifth Int. Conf. Information Fusion* 630–635

Pawlowski, A. M., and P. M. Gerken. 1998. Simulator, Workstation and Data Fusion Components for Onboard/Offboard Multi-Target Multi-Sensor Fusion. *In Proceedings of the 17th Digital Avionics Systems Conference.*

Qt. 2004. User Interface Development Toolkit, Trolltech, Available online via `<www.trolltech.com>` [accessed August 05, 2004]

Srimathveeravalli, G., N. Subramanian, and T. Kesavadas. 2004. A Scenario Generation Tool for DDF Simulation Testbeds. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1698–1705. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Svensson, P., and P. HÖrling. 2003. Building an Information Fusion Demonstrator. *In Proc. Sixth Int. Conf. Information Fusion*

AUTHOR BIOGRAPHIES

CHANDRESH MEHTA is a Master's student in the Department of Mechanical and Aerospace Engineering at the University at Buffalo. He has a BE degree from the University of Mumbai, India. Mehta's research interest includes Geometric Modeling, Computer Aided Product Design, CAE, Finite Element Analysis, Computer Graphics and Simulation. His email address is `<crmehta@buffalo.edu>`

GOVINDARAJAN SRIMATHVEERAVALLI is a Doctoral student in the Department of Mechanical and Aerospace Engineering at the University at Buffalo. He has a BE degree from the University of Madras, India. His email address is `<gks2@eng.buffalo.edu>`

THENKURUSSI KESAVADAS is a professor of Mechanical and Aerospace Engineering at the State University of New York at Buffalo and the founder director of the Virtual Reality lab. He completed his PhD from the Pennsylvania State University in 1995 and M.S from Indian Institute of Technology, Madras in 1988. Prior to joining SUNY Buffalo, he worked as a scientist at the VR center at the Iowa State University. Kesavadas's main areas of research are design of VR based engineering systems, computer aided design, human machine interface, and biomedical simulation. He is a member of ASME and IEEE. His email address is `<kesh@eng.buffalo.edu>`