Status Report for

Planetary Spatial Analyst

NASA NRA NNH06ZDA001N-AISRP Proposal 06-AISRP06-0151
Leslie Keely, PI
June 10, 2008

This is a status report for the project entitled Planetary Spatial Analyst (PSA). This report covers activities from the project inception on October 1, 2007 to June 1, 2008. Originally a three year proposal, PSA was awarded funding for one year and required a revised work statement and budget. At the time of this writing the project is well on track both for completion of work as well as budget.

The revised project focused on two objectives: build a solid connection with the target community and implement a prototype software application that provides 3D visualization and spatial analysis technologies for that community. Progress has been made for both of these objectives.

Connecting with the Community

This objective involves creating the connection to the community that will ultimately use PSA, one of planetary scientists and mission engineers. This connection provides important input as to requirements and usage and was addressed through involvement in 4 collaborations:

1. *Discussions with Ames planetary scientists regarding Mars landing site selection.* An advisory meeting for PSA was held in December 2007. This meeting resulted in a list of requirements for Mars landing site analysis.
    1.1. Easy access to data repositories such as the Planetary Data System (PDS). Scientists stated that many data repositories are difficult to use and it is also difficult to retrieve raw data.
    1.2. Geo-referencing feature. Many data sets are poorly geo-referenced and tools are needed for this purpose.
    1.3. Geo-processing feature. A geo-processing capability for computation and comparison of geo-spatial data was described to attendees and received a strong approval.
2. *Automated Data Assimilation and Flight Planning for Multi-Platform Observation Missions (ADMFP)*. This project was funded by AIST for the 2007 calendar year. It demonstrated automated flight planning using way points mined from data collected during the INTEX-B project. Data sets included atmospheric models, special use airspace regions, and remotely sensed satellite observations. The resulting plan was presented within a 3D visual context consisting of the geographical area of the

mission (the Gulf of Mexico), the satellite overpasses, the special use airspace areas, the atmospheric model, and the candidate way points. In this project the prototype was used to accurately combine data from many different and disparate sources (e.g. simulated vs. observed, in situ vs. remotely sensed, various different scales and projections) and produce a coherent visualization.

3. *ATHLETE's Foot project.* This is a joint project between JPL and Ames and funded by Human Robotic Systems. It is investigating footfall planning for the ATHLETE robot, a large 6 legged rover designed for lunar exploration. This research uses 3D terrain models constructed from stereo images to determine the rover's foot placement. PSA provides a visualization component for the terrain and the robot motion, a ghost robot for plan preview and visual joint force and torque indicators. Input arrives in the form of telemetry from a simulation or the robot itself. For this project, Mercator has been integrated into an existing software application called AthleteWorkbench. This application is being used this summer in a multi-center robotic field test at a lunar analog site in Moses Lake, Washington.
4. *Phoenix Lander Mission to the Martian North Polar Region.* This mission provides PSA with substantial feedback and input from the target community. It is used by scientists to view 3D terrain models acquired by the lander's stereo camera, for shadow studies, and measurements. A very important goal achieved in this mission is that the scientists themselves are using the visualization software instead of a specially designated operator. Also, the software is deployed on the scientists' desktop or laptop instead of a specially configured machine.

Each of the projects listed above have contributed funding to cover the cost of custom software development, travel, and operations. Each project also brings a unique perspective to the PSA development effort with insights from planetary scientists as well as robotic and instrument engineers. The first two introduce PSA to large scale data sets with the second two involving smaller human and robot scale work. Additionally, the projects cover three environments, those of the Earth, Mars, and the Moon.

Prototype Software Application

The revised version of the proposal focused on Mars landing site selection and described a prototype with the following features:

1. Visual data access methods to PDS and the local file system.
2. Visualization of 3D models featuring maps of elevation, slope, and shadow.
3. Data representation in global and local scales.
4. Simple data extraction and measurement tools.
5. Simple annotation and drawing tools.
6. Persistence of software state.

Items 4 and 5 would be de-scoped if necessary.

Most of the these features have been implemented. The primary remaining features are item 1, visual data access, and item 6, persistence. As might be expected, interaction with users reveals many new use cases and requirements. This input sometimes caused a slight re-direction in development priorities but at the same time produced better focused software. The PSA prototype is named Mercator, after Gerardus Mercator the Flemish cartographer who developed the first conventional cylindrical map projection used for navigation and exploration. Development of Mercator is described next.

Design Architecture

Mercator was initially inspired by previous visualization software developed in the Intelligent Systems Division. The Intelligent Robotics Group (IRG) has produced a number of applications used for robotic visualization and simulation and for operations in the MPL and MER missions.

A distinct design philosophy helps in decision making and design direction. The philosophy behind the Mercator design incorporates several ideas.

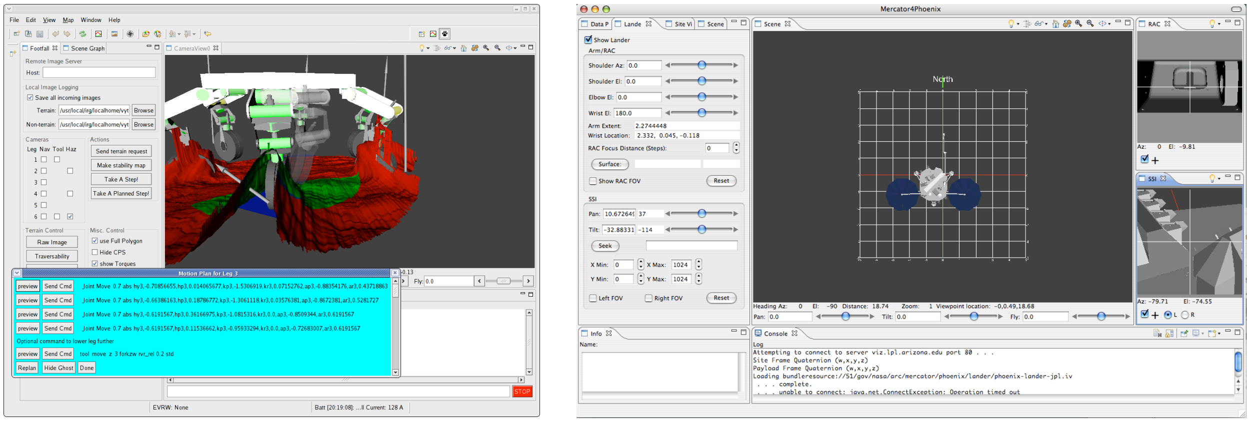
Desktop or laptop deployment: In the past, 3D visualization required specialized expensive equipment or specialized graphics knowledge. The improvements in consumer graphics boards over the last few years makes it now possible to run an advanced graphics application on an average desktop computer. Additionally, the Mercator core was designed and implemented using Java, which is built on newer cross-platform compiler technology, provides many utilities, and is easy to install. Mercator is intended to be easily installed and used much like one would MS Word.

Extendable and interoperable: Basic desktop use should not preclude advanced techniques or new features and interoperation with other mission software. The Mercator design is object-oriented with its foundation built on the concept of a scene graph. The scene graph consists of scene objects which have behaviors. Mercator can be extended by adding new object types and/or new behaviors to a type. Additionally, an object might have the capability to communicate with other mission software or third party software such as IDL or Matlab.

Stand alone application and a software library. Parts of Mercator should be easily incorporated into other applications and should not be difficult for developers to work with. Mercator employs a design feature called a “facade”. In object-oriented software, a facade can provide a simple API and hide details that the developer doesn’t need to know. This simplifies the process of incorporating parts of Mercator into other applications.

Configurable: Extensibility can lead to large and complex software, unavoidable in a richly featured product. Mercator configurability is provided by its plugin architecture based on the Eclipse software framework. A plugin provides a set of features and may be configured into a version of Mercator for a specific mission or field test. Limiting the software to only necessary plugins makes it more manageable by reducing the complexity of installation and training.

Customizable: Mission scientists and instrument engineers are usually extremely busy with mission particulars and don't have time to learn a large complicated application. A simplified, mission specific user interface can help. The facade capability in Mercator can be used in building custom mission user interface plugins. Additionally, a useful feature of the Eclipse framework allows application code to embed web pages and interact with them directly. This capability makes it possible for web page designers to implement a custom user interface for Mercator.



```
Joint Move 0.7 abs hy3,-0.70856655,hp3,0.014065677,kp3,-1.5306919,kr3,0.07152762,ap3,-0.88354176,ar3,0.43718863
Joint Move 0.7 abs hy3,-0.66386163,hp3,0.18786772,kp3,-1.3061118,kr3,0.03576381,ap3,-0.8672381,ar3,0.5281727
Joint Move 0.7 abs hy3,-0.6191567,hp3,0.36166975,kp3,-1.0815316,kr3,0.0,ap3,-0.8509344,ar3,0.6191567
Joint Move 0.7 abs hy3,-0.6191567,hp3,0.11536662,kp3,-0.95933294,kr3,0.0,ap3,-0.72683007,ar3,0.6191567
tool move z 3 forkzw rvr_rel 0.2 std
```

```
Attempting to connect to server viz.lpl.arizona.edu port 80 . . .
Site Frame Quaternion (w,x,y,z)
Payload Frame Quaternion (w,x,y,z)
Loading bundleresource://51/gov/nasa/arc/mercator/phoenix/lander/phoenix-lander-jpl.iv
 . . . complete.
 . . . unable to connect: java.net.ConnectException: Operation timed out
```

Views of Athlete Workbench (left) and the Phoenix configuration of Mercator (right). Each has a set of custom panels specific to its purpose but also builds on basic Mercator functionality.

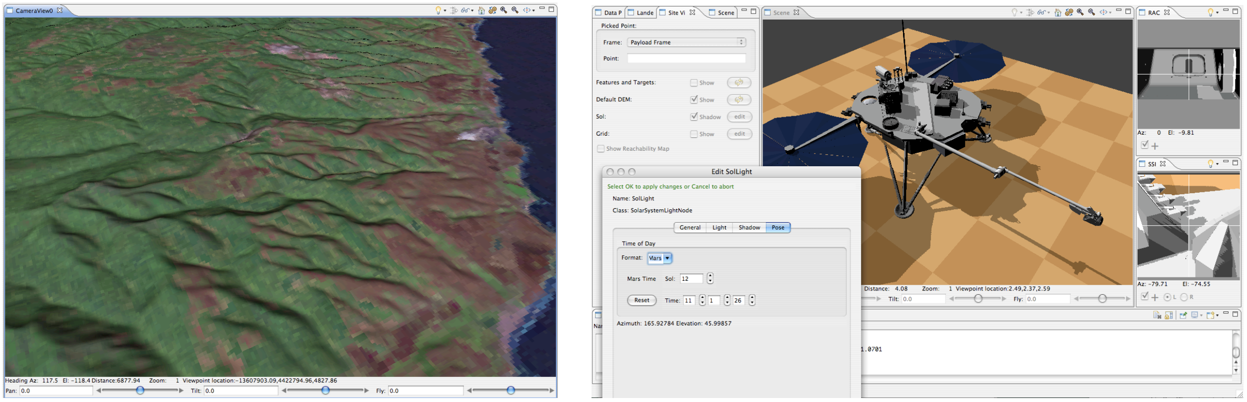
Features and Design Decisions

This section describes some of the major user requirements of Mercator and the design decisions made for the implementation of those requirements. Additionally, the Mercator design tries to address a number of known issues both in interaction and visualization.

Mission software framework: A primary requirement of Mercator was deployment within Ensemble, a multi-center mission software framework based on the Eclipse platform. Mercator employs an open source graphics engine called Java Monkey Engine (jME) for high level 3D rendering. jME is oriented to the gaming market so it was necessary to adapt this software library to scientific visualization and Eclipse. Experiences from this adaptation effort were presented to the developers of jME and will be leveraged in the next release.

3D Models: The main purpose of Mercator is to explore 3D geo-spatial models. Terrain models generally come in two flavors, triangulated irregular network (TIN) and digital elevation model (DEM). Mercator loads and displays both. DEMs can be very large so there is an option to reduce the size. Mercator also loads and displays CAD models from various formats. Model files can be accessed from the local file system or a web server.

Shadows and Illumination: Mars missions and landing site analysis require the capability to determine the light direction of the sun and the shadows it makes. Mercator employs the JPL NAIF SPICE software to compute the direction vector of the light for a given location and time and can display changes in lighting and shadows in real time. There is an option of using Mars time to set the sun position. A mission epoch must be provided. This feature is being used in the Phoenix mission to determine when and where to dig.



Heading Az: 117.5 El: -118.4 Distance:6877.94 Zoom: 1 Viewpoint location:-13607903.09,4422794.96,4827.86
Pan: 0.0 Tilt: 0.0 Fly: 0.0

Picked Point:
Frame: Payload Frame
Point:
Features and Targets: ☐ Show
Default DEM: ☑ Show
Sol: ☑ Shadow
Grid: ☐ Show
☐ Show Reachability Map

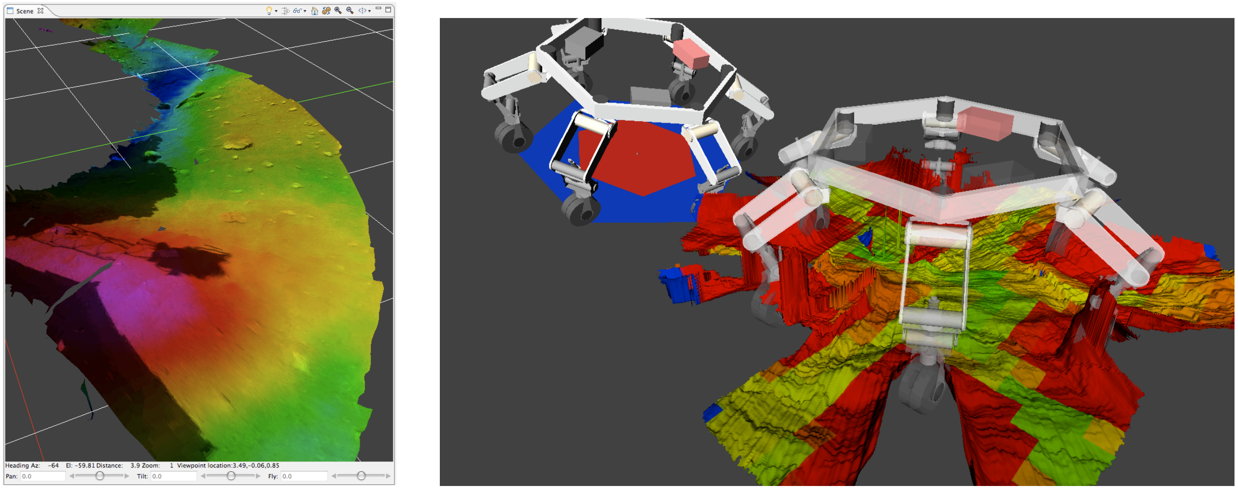
Edit SolLight
Select OK to apply changes or Cancel to abort
Name: SolLight
Class: SolarSystemLightNode
Time of Day
Format: Mars
Mars Time Sol: 12
Time: 11 1 26
Azimuth: 165.92784 Elevation: 45.99857

Distance: 4.08 Zoom: 1 Viewpoint location:2.49,2.37,2.59
Tilt: 0.0 Fly: 0.0
Az: 0 El: -9.81
Az: -79.71 El: -74.55
1.0701

Views of a DEM (left) of the California coastline north of Santa Cruz (courtesy of JPL's On Earth WMS server) and shadows of the Phoenix lander (left).

Visualization of Elevation and Slope: Mercator provides the capability for creating elevation and slope maps for a terrain surface. Several default color ramps are provided or the user can create a new one.

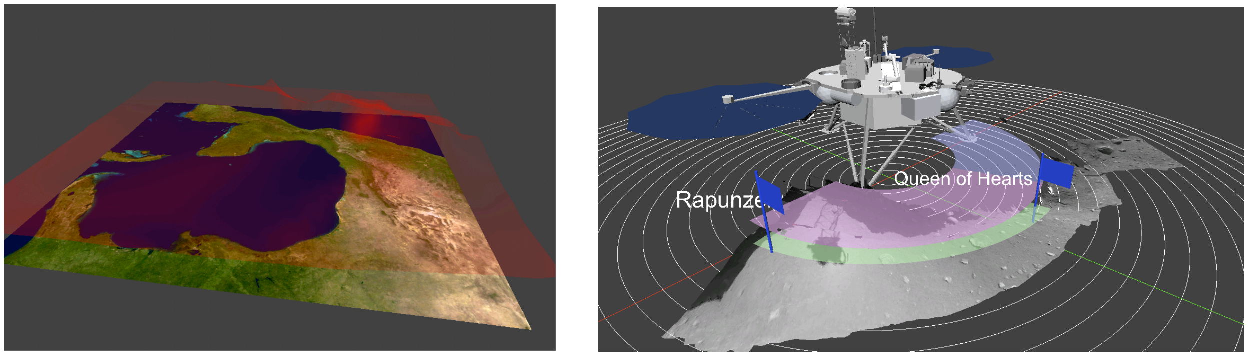
Transparency: Mercator provides the capability to set the transparency of an object. This feature is used to create a ghost model for plan previewing in the ATHLETE's Foot project.



Heading Az: -64 El: -59.81 Distance: 3.9 Zoom: 1 Viewpoint location:3.49,-0.06,0.85
Pan: 0.0 Tilt: 0.0 Fly: 0.0

Views of the Phoenix workspace colored by elevation (left) and a ghost of ATHLETE (right).

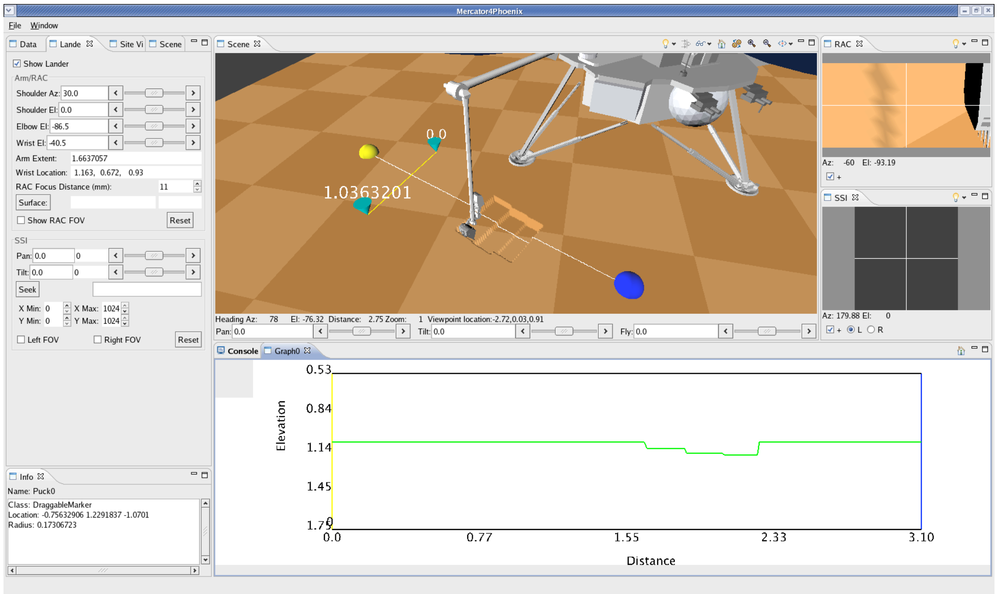
Grids and Annotation: Mercator provides numbered rectangular and radial grids. Additionally, markers of various shapes, sizes and colors may be placed anywhere in the scene. Text can be added and the markers can be turned on and off. The markers can also be moved, scaled and rotated.



Views of one level of CO data from an atmospheric model (left) and the Phoenix workspace with a radial grid, and annotated markers (right).

Measurement: Mercator provides linear, area, and volume measuring tools. The volume and area are estimations. The linear measure can be a single line segment or a poly-line. All measuring tools can be adjusted after they are created. For DEMs, a trenching option is available for the linear measuring tool. This feature was used for Phoenix.

Profiling: DEMs can be profiled and the profile viewed in a graph. A Y-axis reverse option was added for robotic (vehicle) coordinate systems where positive Z values point down.



☑ Show Lander

Shoulder Az: 30.0

Shoulder El: 0.0

Elbow El: -86.5

Wrist El: -40.5

Arm Extent: 1.6637057

Wrist Location: 1.163, 0.672, 0.93

RAC Focus Distance (mm): 11

☐ Show RAC FOV

Pan: 0.0 0

Tilt: 0.0 0

X Min: 0 X Max: 1024

Y Min: 0 Y Max: 1024

☐ Left FOV ☐ Right FOV

Name: Puck0

Class: DraggableMarker
Location: -0.75632906 1.2291837 -1.0701
Radius: 0.17306723

Heading Az: 78 El: -76.32 Distance: 2.75 Zoom: 1 Viewpoint location:-2.72,0.03,0.91

Pan: 0.0 Tilt: 0.0 Fly: 0.0

Az: -60 El: -93.19

☑ +

Az: 179.88 El: 0

☑ + ● L ○ R

View of Mercator with a linear measurement tool and trenches. A profile is also visible with its graph.

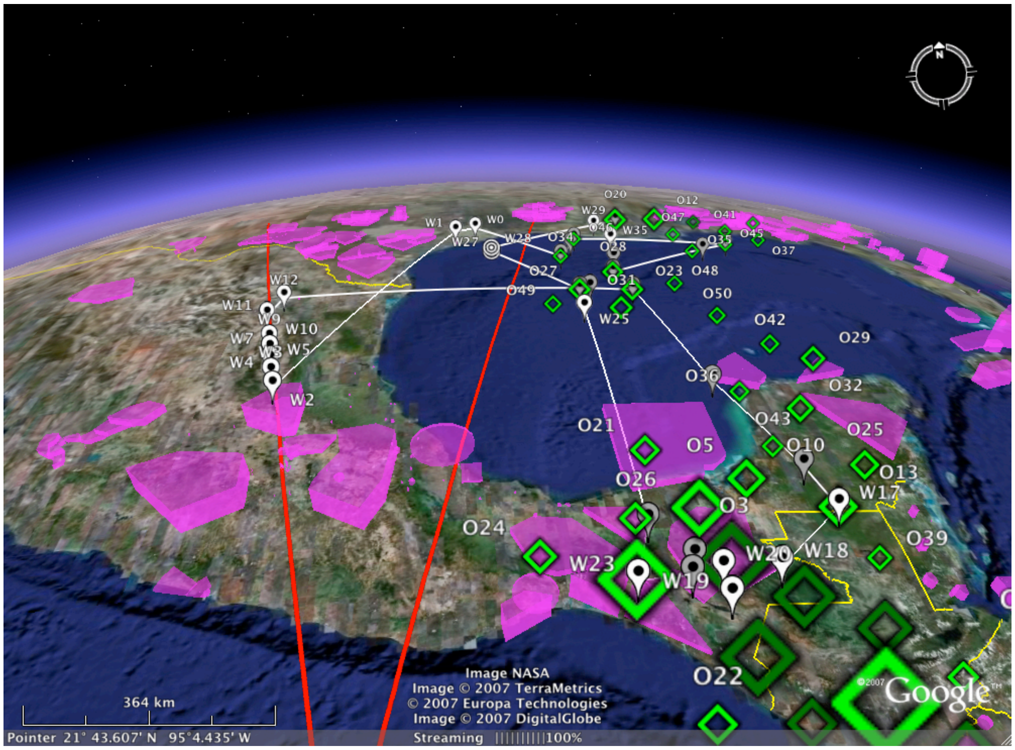
Manipulation of CAD model parts: Mercator maintains a set of software classes for manipulating joints and positions of robot or instrument parts. These classes can be used to point a camera or other instrument or rotate a robot joint. This feature is used in the ATHLETE's Foot project and the Phoenix mission.

Other Visualization Techniques: Mercator can visualize way points and paths such as those designated for aircraft or satellites. Mercator also provides visualization of 3D extruded volumes such as special use air spaces, and atmospheric surface data sets. These visualization techniques were used in the ADMFP project.

Scene Interaction: It is well known that users can easily get lost in a 3D virtual world. Several interaction methods that could minimize this problem have been implemented:

- *Sticky object dragging*. Three dimensional movement in a virtual world can be difficult. This is usually due to lack of the depth cues and feedback humans enjoy in the real world. Mercator is experimenting with a technique for grounding the user. If a scene object is tagged as "draggable", it may be selected and moved with the cursor. The dragged object will "stick" to the scene object underneath it at a point in direct line with the cursor. This technique is currently working well with the profile and measuring tools.
- *Minimal modality*. Drawing programs typically use multiple modes for drawing and editing a view. It is easy to get confused and lost if one forgets one's mode. So far, Mercator is limited to two modes, a default viewing mode and a dragging mode. In viewing mode, mouse movement allows for rotation, translation, and scale of the 3D scene. In dragging mode, movement of the cursor with the left mouse button moves the currently selected draggable scene object. There is no other difference with viewing mode. This approach may require a few more modeless interactions to complete a task but could be more intuitive; more like manipulating an object held in one's hands.
- *Multiple ways to access scene objects.* The user may access a scene object directly from the 3D scene view. If the object is not visible, the user can go to the scene graph panel. This panel is a tree view of the scene graph by object name. The user may bring a given object to the center of the 3D scene view by selecting the seek option.
- *Context Menu*. Different types of objects have different behaviors. For example, a linear measuring tool can provide a trench in a DEM but not a TIN. Mercator employs a context menu in the scene view and in the scene graph tree view. The context menu is invoked by clicking the right mouse button on the desired scene object. The menu displays what behaviors the object provides.
- *Automatic lighting*: Initially, a model loaded from a file will be dark without a light source such as the Sun. Mercator provides three default light options - headlight, skylight, and off. The headlight operates as if the user were wearing a head lamp and the skylight maintains a light directly from above the center of rotation.
- Automatic centering: Many models are not centered at the origin. This can cause confusion for users unfamiliar with the model. Mercator will automatically center the scene when the first model is loaded. An auto-centering button is also provided.

Sharing and Collaboration: Google Earth is a well known software application for geo-spatial visualization. Mercator can save a scene into KML format for ingestion into Google Earth. This feature was used for the ADMFP project.



View of KML result for ADMFP in Google Earth.

Results

The PSA project created a 3D visualization and spatial analysis desktop application for the planetary science community. In the last 8 months Mercator has evolved to a comprehensive tool and has been used for 3 projects ranging from automated flight planning to robotic manipulation to a Mars lander mission. These projects have provided a rigorous testbed for the Mercator design and indicate that there is a place for 3D spatial analysis software in missions at NASA. Interaction from 4 collaborations has provided a rich set of requirements and use cases from which to develop. Users have shown a strong interest in this tool and are inspired to new feature requests. They want to use it to set up instruments and try "what if" scenarios as well and for analysis and measurement.

Additionally, technologies currently only used for games have been harnessed to create a desktop scientific environment. Challenges exist in the combination of many different data sources and techniques into one environment, and in the interaction with that environment. However, the advantages appear to be worth every bit of the effort.

Budget and Work Breakdown

All of the budget for PSA to date has been used for salary. PI Keely performed all software development and Co-I Beyer performed Mercator testing as well as organized the interaction with the community. Dr. David Lees implemented custom data access panels for Phoenix. Dr. Laurence Edwards provided 3D surface reconstruction for Phoenix using the Ames Stereo Pipeline.

For the remainder of FY ‘08, development work will focus on items 1 and 6 of the revised PSA requirements listed above. Use of Ames WorldWind is planned for item 1. Additional work will include further testing of current capabilities and support for the projects that are still ongoing: Phoenix and ATHLETE’s Foot. Additionally, Dr. Beyer will use Mercator to examine high-resolution digital terrain models (DTM), created from HiRISE stereo images, exercising its capabilities for handling orbital data.

Future

A follow-on proposal is planned for the AISRP-2008 call and use of Mercator has been discussed for two additional SMD proposals as well as the MSL mission. Additionally, Mercator will participate in future lunar analog field tests held by the IRG.

Now that the software platform exists the PSA team is very interested in adding features.

- Further development of annotation capabilities, e.g. user settable colors, line thickness, shape, etc.
- Better geometry tools, e.g. triangulation, interpolation, etc.
- Better shadow and lighting using newer techniques.
- Level of detail techniques to handle larger terrains.
- Geo-processing tools.
- The geo-referencing tool specifically called out by members of the Mars landing site advisory meeting.
- More geologic and atmospheric visualization techniques.
- Plugins for communication with commonly used software such as IDL, Matlab, and ISIS.
- Time-lapse.
- Collaboration with other AISR projects.
- Better 3D user interaction techniques possibly with 3D actuators.

These are a few of the possibilities and there will be many more requested by inspired users. As new innovations and techniques appear, an attempt will be made to investigate and incorporate those that show promise.

Acknowledgements

The following people provided advice, support, patience, and time for this project and deserve a great deal of gratitude. It would not have succeeded without them.

Dr. Ross Beyer, PSA Co-I and HiRISE and LROC science team member
Dr. Laurence Edwards - MRO Co-I (CTX) and lead of the IRG visualization team.
Dr. David Lees - member of the IRG visualization and field test teams.
Dr. Carol Stoker - Phoenix mission Co-I and lead of the Phoenix bio-habitability theme group.
Vytas Sunspiral - lead of ATHLETE's Foot project and member of the IRG field test team.