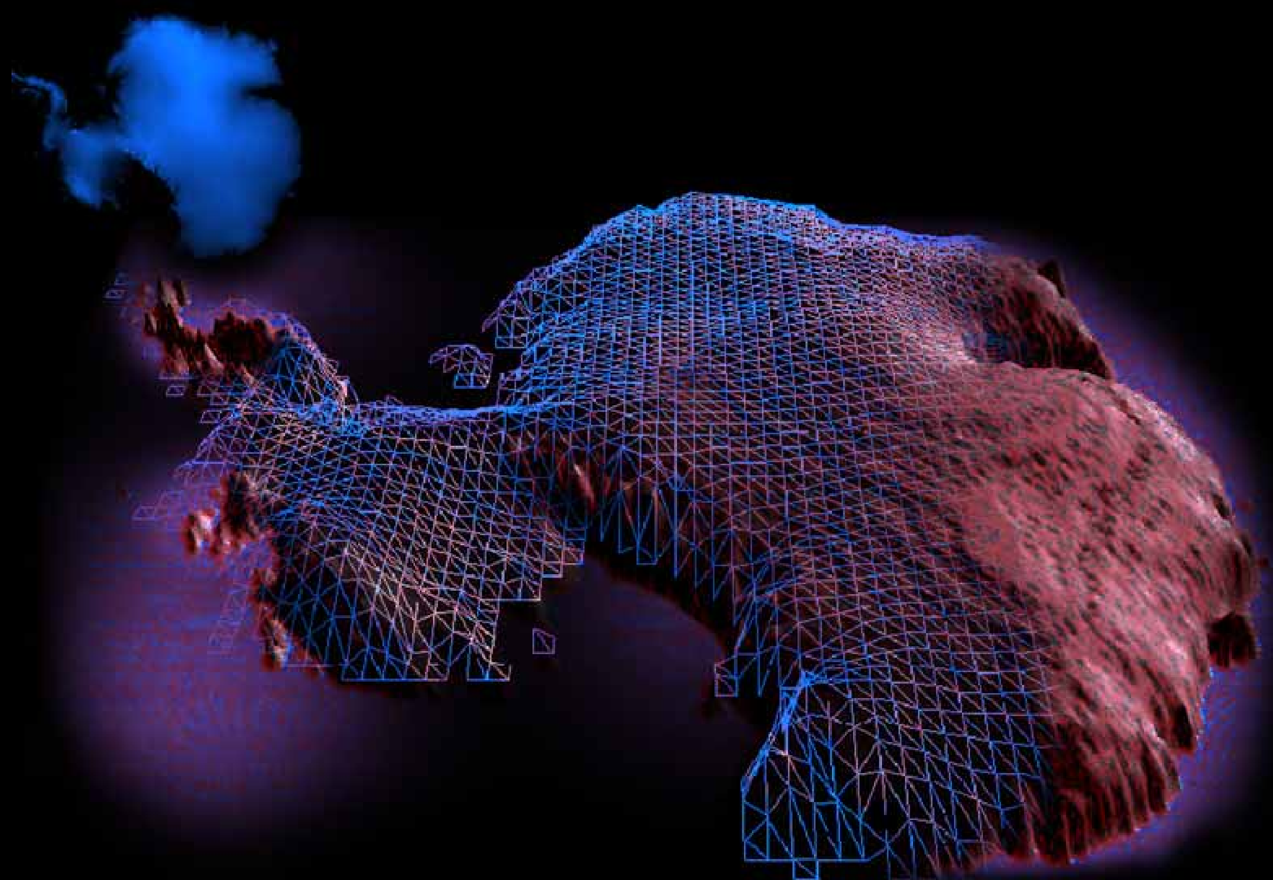
TerraForm*3D*

Project Proposal



Trent D'Hooge

Heather Jeffcott

Craig Post

Deborah Lee Soltesz

USGS/Plasma Works NAU Capstone Design Project

Table of Contents

Table of Figures

Problem Statement

The United States Geological Survey's Mini Image Processing System (*USGS MIPS*) is a fully functional digital image processing software package. MIPS has been ported to the Compaq Alpha under True64 UNIX and Open VMS, Silicon Graphics under IRIX64, and Sun under Solaris. It has the ability to process and create visualization products using a variety of remotely sensed data including Landsat MSS and TM, SPOT, AVHRR, RADAR, SONAR, digitized aerial photographs, digital elevation models, and geophysical data. While the system in general is very powerful, it has several weaknesses:

1) The terrain modeling capabilities lack an interactive interface.
2) Many of the features available in contemporary terrain modeling packages such as flybys[1] of the terrain, and easy rotation of the image.
3) An intuitive user interface for linking multiple programs together to create a final product.

The second sponsor company, Plasma Works, is currently working on a game, named *Storm Saga*, which requires a three-dimensional object rendering engine (*3D engine*). The project is in its mid-design phase and Plasma Works has been planning to license a third party 3D engine. However, the company is interested in an analysis of the value and feasibility of the system and in having experimental spherical coordinate 3D technology developed into a rendering engine. To the knowledge of Plasma Works, a spherical coordinate system has not been implemented in the commercial world. A few examples of why people have not created a spherical 3D engine are:

1) Mathematicians have defined matrices to deal with Cartesian coordinates for a while so all matrix transformations are done using Cartesian coordinates. Cartesian coordinates are not necessary, but in order to use spherical coordinates you would have to rewrite the matrices.
2) Cartesian coordinates are more intuitive and easy to use. People are used to thinking in x,y,z locations in space, rather than distance and angle from the origin.
3) Translations are difficult with spherical coordinates. It involves a lot of trigonometry to translate a spherical coordinate compared to the simple addition of the Cartesian coordinates.

The USGS Astrogeology team and the support organization Office of Program Support (*OPS*) at the Flagstaff Field Center are moving away from the high-end UNIX workstations and servers towards Intel based Linux machines. As Intel based machines have become faster and more powerful, it has become feasible to use them as workstations and servers, decreasing both the cost of the system and the cost of maintaining the hardware and operating system. Bu, Intel systems still lag behind the high-end systems. Consequently, there is interest in using clusters to recapture the lost processing power. Utilization of multiprocessor clusters will allow computationally intensive processes, such as rendering three-dimensional terrain models, to be finished more quickly and efficiently. Currently, in-house image processing software systems do not take advantage of the parallel processing potential of clustered or multiprocessor machines.

Terrain Modeling

Stuart C. Sides of the U.S. Geological Survey (*USGS*) has requested a terrain modeling package be built to supercede the existing labor intensive terrain rendering capabilities in the USGS MIPS image processing system. The package will enable users to interact with a digital elevation model as a method of setting parameters for rendering high quality perspective views, create fly-by movies, and export a model into a portable format for sharing or web distribution. The interactive interface will increase the productivity of the users of the package. A variety of available output formats will allow the users to create visualization products without needing to go to a third-party package, decreasing costs by saving on labor and third-party software. Because of the small user base, USGS MIPS software can be upgraded with new features quickly based on as little as one user request, an important feature not offered by third-party vendors.

3-Dimensional Screen Rendering

Abe Pralle of Plasma Works has requested a real-time 3D engine for its upcoming Storm Saga game. The game is currently being implemented in 2D form for testing, but the final product requires the flexibility that a 3D engine provides in order to be competitive in the market. It will be developed and tested as a 3D rendering component of the USGS terrain modeling package, which requires 3D screen and file rendering capabilities and the capability to take advantage of 3D video hardware acceleration.

Parallelization of Image Processing Procedures

Margaret Johnson of the USGS OPS has requested an inquiry into the advantages, feasibility, and required resources for the implementation of multiprocessor systems, especially clustered systems, in an image processing production environment. The terrain modeling package will have some processor intensive functionality and will serve as a prototype for parallelized image processing software for the purpose of benchmarking and demonstration of the concept. If successful, clustered systems could replace high-end, high maintenance workstations in the future.

Business Environment

Plasma Works

Plasma Works has a very relaxed business environment. It is currently a privately owned and funded operation with six employees. There are three programmers, two artists, and one designer/story writer. Plasma Works is looking to expand into the 3D engine market and game development market over the next few years.

USGS

The USGS has a team oriented structure, with each functional Division divided into Regional administrative offices, and finally into small specialization Teams. The USGS Flagstaff Field Center has a very relaxed environment with a small population operating under the *open door policy*.

MIPS Administrators and Users

Many software packages, including USGS MIPS, allow the user to have direct access to the software development team, and can request bug fixes, technical and application help, platform support, features and enhancements, and support for new data types. MIPS is an open source package currently supported by a single programmer who depends upon MIPS' encapsulated organization and reusable components to facilitate maintaining the package.

USGS OPS System Administrators

The OPS systems administration team currently supports all computer hardware, software, and network issues at Flagstaff Field Center. They keep up to date on all developments in the computer environment to insure that the users know about any improvements that can help their productivity. Users are given one-on-one, personalized support. Most users access multi-user UNIX-based systems through X-server[2] interfaces running on Windows-based[3] systems. On average, there are four users per system, with two systems devoted to a single user each. The external users of the software developed by the Astrogeology group are on a variety of different types of computers. OPS maintains several different platforms and operating systems for the software developers. Budget is always an important factor, and OPS strives to find low-cost solutions whenever possible.

Scope of Development

3D Engine Solution

The *PlasmaWorks 3D Engine* will be developed for Abe Pralle that will serve as a test of the *Spherical Coordinate vs. Cartesian Coordinate* research ideas. It will contain all of the functionality Plasma Works has requested and be delivered on April 28, 2000. The 3D engine will have an understandable interface and use the latest hardware acceleration provided through the DirectX and OpenGL graphics libraries.

Terrain Modeling Package

Terraform3D will be designed as a functional terrain modeling and visualization package using as a minimum native MIPS image and terrain file types. It will be a replacement for the USGS NUVU[4] and SHADRF[5] programs, and will have the added functionality of real-time flybys and export of flybys as movie frames and VRML[6], valuable formats for illustrations in digital documentation. It will be fully documented, tested, and ready for public distribution as version 1.0 of this package by April 28, 2000. It will make use of a light version of the PlasmaWorks 3D Engine, allowing the source code to be freely distributed. Perspective view output functionality will be parallelized as a demonstration for USGS OPS of the feasibility and advantages of Beowulf clustering[7].

Parallelization

A prototype Beowulf cluster, named *Black Tie and Tails,* will be designed and built for USGS OPS. It will comprise Intel-based systems running the Linux operating system, and be ready for final delivery on or before April 28, 2000. The clustered system will be outfitted with tools for software developers, systems administrators, and system users, and will be fully documented.

Benchmarking

The following summary and reference list shows what products were researched in order to learn the current state of research, development, and available consumer products.

Terrain Modeling and Scientific Visualization

Summary

Feature	Attributes, Description, and Comments
Input file format flexibility	USGS Digital Elevation Models, bathymetry, ASCII, and GeoTIFF, in addition to popular image formats
Output file format flexibility	High resolution renderings and multimedia, i.e., virtual reality and movie formats
Image/texture overlay	Texturize terrain using digitized maps and remotely sensed image data
Color-coding elevations	Preset and user-defined color palettes
Adjustable sun-angle	User definable location of lighting source
Surface rendering options	Diffuse and specular reflection values, anti-aliasing
Real-time flyby	Keyboard, mouse, or joystick input used to virtually navigate/fly through terrain
Multi-resolution screen rendering	Animations (i.e., real-time flybys) happen at low resolution in order to run smoothly; screen image renders at higher resolution when action stops
Data feedback	Query tools for geographic location, elevation, or other values
Multiple views	Along axis, perspective, orthogonal, and plan views
Flight tracks	Ability to record and replay flight tracks

Consumer Software

Rapid Imaging Software: Landform http://www.landform.com/

MetaCreations Bryce http://www.metacreations.com/products/bryce4/

Geomantics Landscape Visualization http://www.geomantics.com/

Download Planet Earth http://www.geocities.com/TimesSquare/1658/earth.html

TruFlite Software http://www.truflite.com/

3DEM Freeware Terrain Visualization and Flyby Animation
http://www.monumental.com/rshorne/3dem.html

Okino Computer Graphics NuGraf rendering system
http://www.okino.com/products.htm

3D Engines, Rendering, and Applications

Summary

Feature	Attributes, Description, and Comments
Full Texturing and Lighting	Standard Texturing, Mip Mapping, Multi-Texturing
Tools	3D Modelers, Animators, Level Designers
Neat Graphic Effects	Lens Flare, Flame Effects, Bill-boarding, Shadows, Colored Lighting, etc.
Model Effects	Deformable Meshes, Realistic Physics, Weighted Vertices
Hardware Acceleration	Direct3D, OpenGL, Glide
Commercial Plug-ins	3D Studio Plug-ins and file compatibility

Consumer Software, Libraries, and Developer's Kits

GIG3DGO 3D modeling software
http://www.linux-magazin.de/ausgabe/1997/12/3DGO/3dgo.html

The 3D Engines List http://cg.cs.tu-berlin.de/~ki/engines.html

Genesis3D Open Source Project http://www.genesis3d.com/

Monolith LithTech 3D game engine http://www.lith.com/lithtech/

Quake III Engine http://www.idsoftware.com

Unreal Engine http://www.unreal.com

Parallelization

Summary

Feature	Attributes, Description, and Comments
Clusters used by many organizations	Used for many different purposes
Uses MPI	There are free distributions of MPI
Code solves specific problem well	Does not work well for other uses
Very little cost	Runs on Linux, a open source operating system
Vast resources of information on the web	It is easy to find a solution to a problem.

Examples of clustered systems, software, and Beowulf projects implemented by various organizations

Beowulf home page. http://www.beowulf.org

Compaq Alpha Parallel Processing http://www.microway.com/links.html

Project Components

Components

The project is divided into three components based on the needs of the clients and the expertise each member brings to the team:

- **3D Engine:** a generic mathematical, display, and rendering library for manipulation of digital representational models of three-dimensional real world objects
- **Terrain Modeling:** a scientific package built around the 3D engine used for the interactive manipulation of a digital elevation model, and creation of terrain visualization products
- **Parallelization:** implementation of a cluster/multiprocessor system and parallelized software for the demonstration of the usefulness of multiple processor systems in the image processing environment

Team

A team of four members has been given the task of the design and implementation of the package. Three of the members will each take responsibility for the three main components of the project and serve as the liaison to the appropriate sponsor, as shown below:

Component	Leader	Sponsor
Terrain Modeling	Deborah Lee Soltesz	Stuart C. Sides
3D Engine	Craig Post	Abe Pralle
Parallel Systems	Trent D'Hooge	Margaret Johnson

Table 1: Project Component Management

The fourth team member, Heather Jeffcott, will serve to support the leaders by performing research, documentation, and coding tasks as needed.

In addition to delivering weekly status reports, incremental results and prototypes, and documentation, the team will demonstrate the final product on 28 April 1999 at the NAU Capstone Engineering Design Conference. An additional presentation and poster session at the USGS will be held on a date to be announced.

Methodology

Due to the nature of the problem, and the lack of experience dealing with the development of a 3D engine and parallelizing image manipulation algorithms, the method of development will be prototyping. This will allow hidden problems to be found early in the preliminary architecture stage, make changes easily, reanalyze the risks and feasibility, and adapt the architecture.

Timeline

Figure 1 shows a overview of the major tasks and deliverables, the time duration required to complete each item by its deadline.

Task Name	Start Date	Effort	Duration	End Date
Documents	**9/30/99**	**111.50d**	**101.50d**	**2/24/00**
Team Standards Document	9/30/99	2.00w	10.00d	10/13/99
Problem Statement	9/30/99	1.00mon	21.75d	10/29/99
Proposal Document	10/29/99	1.00mon	21.75d	12/1/99
Design Document	12/2/99	58.00d	58.00d	2/24/00
Parallel	**9/30/99**	**43.50d**	**43.31d**	**12/2/99**
BT and T Running	9/30/99	1.25mon	27.19d	11/8/99
Image Processing	11/8/99	0.75mon	16.31d	12/2/99
Terrain Modeler	**9/30/99**	**144.00d**	**144.00d**	**4/24/00**
Build Terrain Modeler	9/30/99	144.00d	144.00d	4/24/00
3D Engine -- USGS	**9/30/99**	**144.00d**	**144.00d**	**4/24/00**
Build 3D Engine	9/30/99	144.00d	144.00d	4/24/00
3D Engine -- PlasmaWorks	**9/30/99**	**144.00d**	**144.00d**	**4/24/00**
Build 3D Engine	9/30/99	144.00d	144.00d	4/24/00

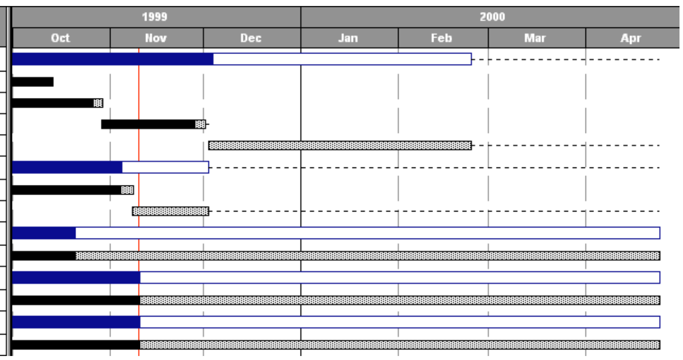


Figure 1: Project Timeline

General System Requirements

This section lists the requirements for the final design gathered from the original project proposals and communications with the clients. Difficulty rating of implementing a requirement is stated in a scale of 1 to 5, where 1 is very difficult and 5 is very easy. Each factor is prioritized on a scale of 1 to 5 as follows:

1) Vital
2) Important
3) Moderately Important
4) Design improvements to be implemented dependent on time and budget constraints
5) Important to the future development and integration of the package

3D Engine

Item #	Requirement	Importance	Difficulty
1.	Spherical and Cartesian coordinate systems (inquiry into feasibility of spherical coordinates)	1	4
2.	Full texturing and lighting effects	3	3
3.	Animation-able Models	2	1
4.	A free floating camera	3	4
5.	Polygon culling	3	3
6.	Direct3D compatibility	1	3
7.	OpenGL compatibility	2	3
8.	Legal agreement for use and ownership of 3D Engine.	1	5
9.	A solid, stable architecture that is easily modifiable	1	2

A Free Floating Camera:

A free floating camera can be positioned at any point in 3-space and set to point in any direction. The screen is rendered from the view of the camera.

Polygon Culling:

Polygon Culling removes polygons from the rendering pipeline if they are out of view of the camera, either by being far away or covered by other objects.

Direct3D/OpenGL Compatibility:

Direct3D and OpenGL are hardware-accelerated rendering APIs.

Spherical and Cartesian Coordinate System:

A spherical coordinate system is one in which vertices are represented by a distance from the origin and two angles representing pitch and yaw. In a Cartesian coordinate system the vertices are represented as x,y,z coordinates.

Explanation:[8] The coordinates used in spherical coordinates are rho (ρ), theta (θ), and phi (ϕ). Rho is the distance from the origin to the point. Theta is the angle between the x-axis and the line connecting the origin and the point. Phi is the angle between the z-axis and the line connecting the origin and the point.

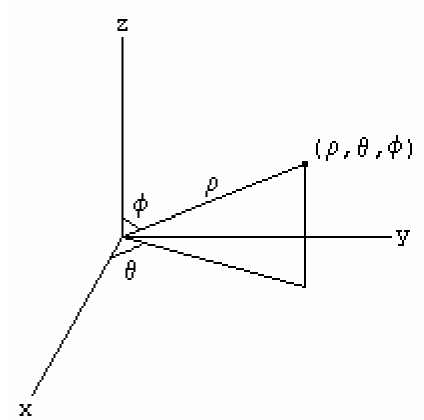


Figure 2: Spherical and Cartesian Coordinate Systems

The following are the relations between Cartesian and spherical coordinates:

From spherical to Cartesian:

$$\rho = \sqrt{x^2 + y^2 + z^2}, \quad \tan\theta = \frac{y}{x}, \quad \tan\phi = \frac{\sqrt{x^2+y^2}}{z}$$

From Cartesian to spherical:

$$x = \rho \sin\phi \cos\theta, \quad y = \rho \sin\phi \sin\theta, \quad z = \rho \cos\phi$$

Terrain Modeling Package

Item #	Requirement	Importance	Difficulty
10.	Platform independence	1	4
11.	Visual, interactive user interface	1	2
12.	Input file flexibility	5	3
13.	Output product/format variety (rendering and multimedia)	2	3
14.	Hardware acceleration (*taking advantage of processing power residing on video cards to free other system resources and produce higher quality screen output*)	1	2
15.	High resolution screen re-rendering on halt of animation	1	1

Parallelization

Item #	Requirement	Importance	Difficulty
16.	Development of programming environment	2	3
17.	Implementation of parallelization in software for benchmarking and proof of feasibility	1	1
18.	Create a Beowulf cluster running Linux	1	2

System Specifications

This section lists the specifications for the final design gathered from the original project proposals and communications with the clients. Difficulty rating of implementing a requirement is stated in a scale of 1 to 5, where 1 is very difficult and 5 is very easy. Each feature is prioritized on a scale of 1 to 5 as follows:

1) Vital
2) Important
3) Moderately Important
4) Design improvements to be implemented dependent on time and budget constraints
5) Important to the future development and integration of the package

Features assigned N/A are either not possible or not feasible because of legal issues.

3D Engine

Capabilities and Additions

Item #	Requirement	Importance	Difficulty
1.	Compible under Windows 95/98	1	4
2.	Direct3d optimized	1	2
3.	Spot lighting	2	3
4.	Bulb lighting	2	3
5.	Directional (sunlight) lighting	2	3
6.	Standard texturing	2	3
7.	Wrapping texturing	2	3
8.	Mapped (tiled) Texturing	2	3
9.	Weighted vertices	3	2
10.	Mesh based model animation	2	2
11.	Keyfame based animations	2	2
12.	Model translation/rotation/scaling	2	5
13.	Generic DrawObjectAt(ScreenCoords) function	3	3
14.	Free floating camera	2	4
15.	Spherical coordinate system	2	3
16.	Cartesian coordinate system	2	4
17.	Object/polygon culling	2	4
18.	Lens flare	5	3
19.	Fire effects	5	3
20.	2D Bill-boarding	4	4
21.	Fogging	4	2
22.	Colored lighting	3	3
23.	Realistic shadows	4	1
24.	Fast shadows	2	2
25.	Panoramic sky texture	2	3

26.	Transparency	4	2
27.	Custom 3D modeler	3	1
28.	Texture editor/applier	3	2
29.	Demonstration movie exhibiting engine abilities	4	2

Client Programmer Interface

Item #	Requirement	Importance	Difficulty
30.	Comprehensive code documentation	1	4
31.	Tutorial and demonstration programs using 3D Engine	1	3
32.	Flexible architecture enabling effects to be added later.	3	2

Terrain Modeling Package

Administration and Programming Environment

Item #	Requirement	Importance	Difficulty
33.	Compiled under Windows 95/98	1	4
34.	Compiled under Windows NT	1	4
35.	Compilable under True64[10] UNIX and Digital[11] UNIX 4.x	1	4
36.	Compilable under SuSE[12] Linux	1	4
37.	Compilable under Solaris[13] UNIX	4	4
38.	Compilable under MacOS[14]	4	4
39.	Compilable under IRIX64[15] UNIX	4	5
40.	Implemented as parallelized software under Beowulf[16]/MPI[17]	1	1
41.	Implemented as serialized software under all platforms	1	4
42.	Documentation: programmer's reference	1	5
43.	OpenGL 3D rendering	1	2
44.	Encapsulated organization	2	4
45.	Reusable components	2	4
46.	C-compatible programming language	3	5
47.	Installing requires no other software investment, such as compilers, but may require minimum hardware configuration	1	5

User Environment

Item #	Requirement	Importance	Difficulty
48.	Real-time manipulation and display of 3D terrain using OpenGL	1	2
49.	Real-time manipulation and display of 3D terrain using software rendering	3	2
50.	Windowing system using platform independent environment (either GLUT[18] or Java)[19]	1	2
51.	Terrain manipulation: rotation around Z-axis (elevation axis)	1	2
52.	Terrain manipulation: rotation around X-axis (tilting)	1	2
53.	Terrain manipulation: rotation around Y-axis (tilting)	1	2
54.	Terrain manipulation: panning	1	2
55.	Terrain manipulation: zooming	1	2
56.	Terrain manipulation: elevation exaggeration	1	2
57.	Terrain visualization: creation of flight path via setting way points, saved as a configuration file	2	3
58.	Terrain visualization: creation of flight path via capturing location information during real-time flight		
59.	Command line mode (software functionality without launching GUI)	3	3
60.	Read very large image files using random file access (instead of reading entire file into memory). In general, image and terrain files are 25 to 50 megabytes, but may be over 1 gigabyte in size.	1	5
61.	Input/Output file type: MIPS native image and DEM* files	1	5
62.	Input/Output file type: Uncompressed Tagged Image File Format (TIFF)	3	5
63.	Input/Output file type: Compressed TIFF	5	1
64.	Output file type: LZW compressed TIFF (compression algorithm not freely licensed)	N/A	N/A
65.	Input/Output file type: Uncompressed Windows Bitmap (BMP)	5	5
66.	Input/Output file type: Joint Picture Expert Group format (JPEG)	5	1
67.	Input file type: CompuServe Graphics Image Format (GIF)	5	3
68.	Output file type: GIF (compression algorithm is not freely licensed)	N/A	N/A
69.	Output product: high resolution perspective view using ray-tracing techniques	1	1
70.	Output product: user defined resolution image sequence (for use with digital video software, such as Apple QuickTime)	2	4
71.	Output product: Virtual Reality Modeling Language (version 2) at user defined detail level	2	4
72.	Output product: VRML version 1 is not efficient for this application because of size constraints	N/A	N/A
73.	Input device: mouse	1	5
74.	Input device: joystick	5	1
75.	Input device: keyboard	1	5
76.	Documentation: installation and user manuals		

Parallelization

Administration and Programming Environment

Item #	Requirement	Importance	Difficulty
77.	X window System	2	4
78.	Secure computer system, keep clustered computers behind a firewall[20].	1	2
79.	Graphical representation of the cluster resources	2	3

User Environment

Item #	Requirement	Importance	Difficulty
80.	A GUI for developing and running LAM in.	3	3
81.	Have GNU[21] Fortran installed	1	2
82.	Ability to use MPI in Fortran	2	3
83.	Have GNU C installed	1	2
84.	Ability to use MPI in C	2	3
85.	Have GNU C++ installed	1	2
86.	Ability to use MPI in C++	2	3
87.	Accessible from other computers	3	4

Use of Technology

3D Engine

The 3D Engine will take advantage of the latest hardware acceleration. This includes hardware texturing, rendering, culling, and lighting. It will be able to access files from either a commercial 3D modeling package or a custom modeling package to be provided with the engine.

In order for the 3D Engine to remain competitive, it must take advantage of as many 3D video cards on the market as possible. This will require testing the 3D Engine on multiple systems with different configurations.

Terrain Modeling Package

The terrain modeling package will take advantage of environments and languages which are platform independent in order to allow all users to have access to the tool. OpenGL is a widely accepted standard supported by a large portion of the video cards available on the market. The GLUT toolkit for programming under OpenGL has been designed and compiled for use under a variety of operating and windowing systems. The alternative package for windowing is Java, which is also available for a variety of platforms. C++ has been chosen for the base programming language for its platform availability, speed as a compiled language, compatibility with C, and object oriented structure. All the above required libraries, interpreters, and compilers are freely available.

Final documentation for the package will be publicly available via the WWW and included in the MIPS documentation on the USGS TerraWeb web server.

Parallelized System

The parallelized part of this system will be used to reduce the time it takes to do image transformations. The parallelization will be done using LAM[22], a freely distributed programming and development environment for MPI.[23] Given that LAM and other MPI implementations are free, the code for this part of the project will be given away freely also.

The software will be targeting schools, government, and other agencies that have limited funds for equipment to do large scale image processing. The use of Linux running on off the shelf hardware and the software being freely distributed will reduce the cost of implementing this project.

Project Deliverables

TO BE COMPLETED: Schedule of due dates for deliverable products, documentation, and presentations.

3D Engine

Item #	Item	Delivered to	Deadline
1.	PlasmaWorks 3D Engine (light version for TerraForm3D)	Deborah Lee Soltesz, Stuart C. Sides, Abe Pralle	TBA
2.	Design documentation	Deborah Lee Soltesz, Stuart C. Sides, Abe Pralle	TBA
3.	Programmers' documentation	Deborah Lee Soltesz, Stuart C. Sides, Abe Pralle	TBA

Item #	Item	Delivered to	Deadline
4.	High level design presentation	Abe Pralle	Dec. 17, 1999
5.	PlasmaWorks 3D Engine (full version) prototype 1	Abe Pralle	TBA
6.	PlasmaWorks 3D Engine, prototype 2	Abe Pralle	TBA
7.	Design presentation	Abe Pralle	TBA
8.	PlasmaWorks 3D Engine, final	Abe Pralle, Capstone Design Conference	April 28, 2000
9.	Design documentation	Abe Pralle	April 28, 2000
10.	As-built design presentation	Abe Pralle, Capstone Design Conference	TBA
11.	Programmers' documentation	Abe Pralle	April 28, 2000

Terrain Modeling Package

Item #	Item	Delivered to	Deadline
12.	High level design presentation	Stuart Sides	Dec. 17, 1999
13.	TerraForm 3D, prototype 1	Stuart Sides	TBA
14.	TerraForm 3D, prototype 2	Stuart Sides	TBA
15.	Design presentation	Stuart Sides	TBA
16.	TerraForm 3D, final	Stuart Sides, Capstone Design Conference	April 28, 2000
17.	Programmer/design documentation	Stuart Sides, Capstone Design Conference	April 28, 2000
18.	Installation documentation	Stuart Sides	April 28, 2000
19.	User documentation	Stuart Sides	April 28, 2000
20.	As-built design presentation	Stuart Sides, Capstone Design Conference	April 28, 2000
21.	TerraForm 3D, final	USGS	May 2000
22.	Programmer/design documentation	USGS	May 2000
23.	As-built design presentation	USGS	May 2000

Parallelization

Item #	Item	Delivered to	Deadline
24.	Black Tie and Tails Beowulf cluster, incremental	Margaret Johnson	Dec. 17, 1999
25.	Parallel image processing code demonstration	Margaret Johnson and Stuart Sides	Dec. 10, 1999
26.	High level design presentation	Margaret Johnson	Dec. 17, 1999
27.	Black Tie and Tails Beowulf cluster, incremental	Margaret Johnson	TBA
28.	Design presentation	Margaret Johnson	TBA
29.	Black Tie and Tails Beowulf cluster, final	Margaret Johnson, Capstone Design Conference	April 28, 2000
30.	Administrators/design documentation	Margaret Johnson, Capstone Design Conference	April 28, 2000
31.	Programmer/User documentation	Margaret Johnson	TBA
32.	As-built design presentation	Margaret Johnson, Capstone Design Conference	April 28, 2000
33.	TerraForm3D benchmarking and recommendations	Margaret Johnson, Capstone Design Conference	April 28, 2000
34.	Black Tie and Tails Beowulf cluster, final	USGS	May 2000
35.	Administrators/design documentation	USGS	May 2000
36.	As-built design presentation	USGS	May 2000
37.	TerraForm3D benchmarking and recommendations	USGS	May 2000

Resources

The following languages, packages, libraries, and file formats will be used for the project.

Terrain Modeling Package

Tools and Standards

Item #	Tool	Format, Software, Hardware, or System
1.	User documentation	Adobe Portable Document Format (PDF); HyperText Markup Language (HTML)
2.	Programmer documentation	Adobe Portable Document Format (PDF); HyperText Markup Language (HTML)
3.	Code documentation	Modified USGS MIPS file headers, which require specific disclaimers and legal statements in addition to standard file information
4.	Revision tracking	CVS (Concurrent Versions System)
5.	Programming language	C++
6.	Programming interface to native windowing environment	GLUT or Java
7.	Video hardware acceleration	OpenGL
8.	Software design	Simply Objects Modeler supplemented with Rational Rose Software as needed

Online Resources

Terrain Modeling Schemes
http://www.cl.cam.ac.uk/users/pjcb2/firstyearrep/node7.html

Interactive viewing of 3D terrain models using VRML
http://www.dhpc.adelaide.edu.au/reports/043/html/index.htm

Terrain Visualization http://cs.ru.ac.za/homes/cssb/terrtalk/terrain.html

Parallelized System

Tools and Standards

Item #	Tool	Format, Software, Hardware, or System
9.	User Documentation	Adobe Portable Document Format (PDF)
10.	Programmer Documentation	Adobe Portable Document Format (PDF)
11.	Miscellaneous Documentation	ASCII, HTML, or web browser-compatible formats as required
12.	Cluster software	Beowulf
13.	Parallelization library	MPI

Online Resources

Information about the Message-Passing Interface API and the *Cluster Tux* Beowulf project

Message Passing Interface standards page
http://www-unix.mcs.anl.gov/MPI/index.html

3D rendering and clusters

Photorealistic Rendering by a Monte Carlo Method
http://www.cacr.caltech.edu/research/Beowulf/rendering.html

Porting a 3D Renderer for Spherical Surfaces to Beowulf
http://nccsinfo.gsfc.nasa.gov/VSEP96/vsephtml_corr2/ramanathan.html

Collage cluster, rendering, and Active Mural display
http://www-fp.mcs.anl.gov/collage/

POVBench: POVRay Benchmarking http://www.haveland.com/povbench/

3D Engine

Tools and Standards

Item #	Tool	Format, Software, Hardware, or System
14.	Programmer Documentation	Adobe Portable Document Format (PDF)
15.	Code Documentation	Visual Studio 5.0 (C++) or vi
16.	Revision Tracking	Visual Studio 5.0 (C++)
17.	Commercial 3D Modeler	Animation Master
18.	Image Editor	Micrografx Picture Publisher

Online Resources

Advanced Graphics Programming with OpenGL
http://www.sgi.com/software/opengl/advanced98/notes/notes.html

SGI's OpenGL Homepage http://www.sgi.com/software/opengl/

Links to visualization topics http://www.swin.edu.au/astronomy/pbourke/

3D Modeling Research Papers http://wwwmath.uni-muenster.de/cs/u/mam/papers/

Advanced Topics in Computer Graphics
http://www.math.tau.ac.il/~daniel/Graphics/adv_graph.html

Benefits to the Sponsors

USGS MIPS Team

The terrain modeling software will give USGS MIPS users and maintainers the following benefits:

- decrease the training and learning time required to become proficient at the task of creating 3D visualization products
- save labor hours by decreasing the time required to create 3D visualization products
- eliminate the need for third party software, consequently saving money on the procurement and maintenance of the software, eliminating training required to learn the software, and reducing labor hours used moving intermediate products between incompatible packages
- maintain the software in-house, giving users the ability to request and receive new features in a timely manner
- create multimedia visualization products quickly and easily for presentations, web pages, and other digital documentation

Plasma Works

The 3D engine allows both Plasma Works and US/GS to evaluate new untested methods for their current projects for little or no cost. This project component will test out the feasibility of a spherical coordinate system for animatable model design. It will also test both Direct3D and OpenGL rendering pipelines for increased compatibility. Finally, the 3D Engine will allow Plasma Works to save both money and man hours on the development of a sophisticated piece of graphics software.

USGS Astrogeology and OPS

A clustered system, specifically a Beowulf cluster, and parallelized software to be created for the USGS have the following benefits:

- cost of creating a Beowulf cluster is much less than buying a super-computer or high-end workstation
- the Linux operating system and supporting software for the cluster are free of cost, drastically decreasing software maintenance costs
- done in open-source, allowing other people to access source code and make changes that would improve the software.
- recognition by other institutions

Risk Analysis

After analyzing the problem at hand, the team has come up with the following risks and potential solutions:

Risks	Risk Analysis/Mitigation
The spherical coordinate system could prove too slow and unwieldy.	There may be some mathematical work-arounds to speed up the spherical coordinate computations.
Porting among different platforms and operating systems could prove difficult.	The code must be written in a way to minimize operating system specific commands, and be ported and tested at every major step to strengthen the code and eliminate problems.
3D engines are notoriously difficult and time consuming.	Concentrate efforts and achieve a solid early design to minimize the risks of not finishing such a large project as a 3D engine.
Hardware Failure, computers in the cluster could fail.	The cluster can lose a few computers and still prove the concept. There is also the possibility of creating the operating system on a CD and running these “CD” systems on other borrowed computers without affecting what they currently have installed.
Lack of experience in parallelization: no one on the project has been on a project like this.	Currently three of the members are taking a course in parallel computing, which will give them more experience and knowledge.
Failure to find a good way to parallelize the problem due to lack of experience or the problem as presented cannot be parallelized	Talking to the team that developed MIPS, a suitable way to run parts of the program in parallel should be feasible. Parallelization of image processing applications and 3-dimensional rendering is well documented, and should provide a valuable reference in the design and development.

Design

Component Interaction Overview

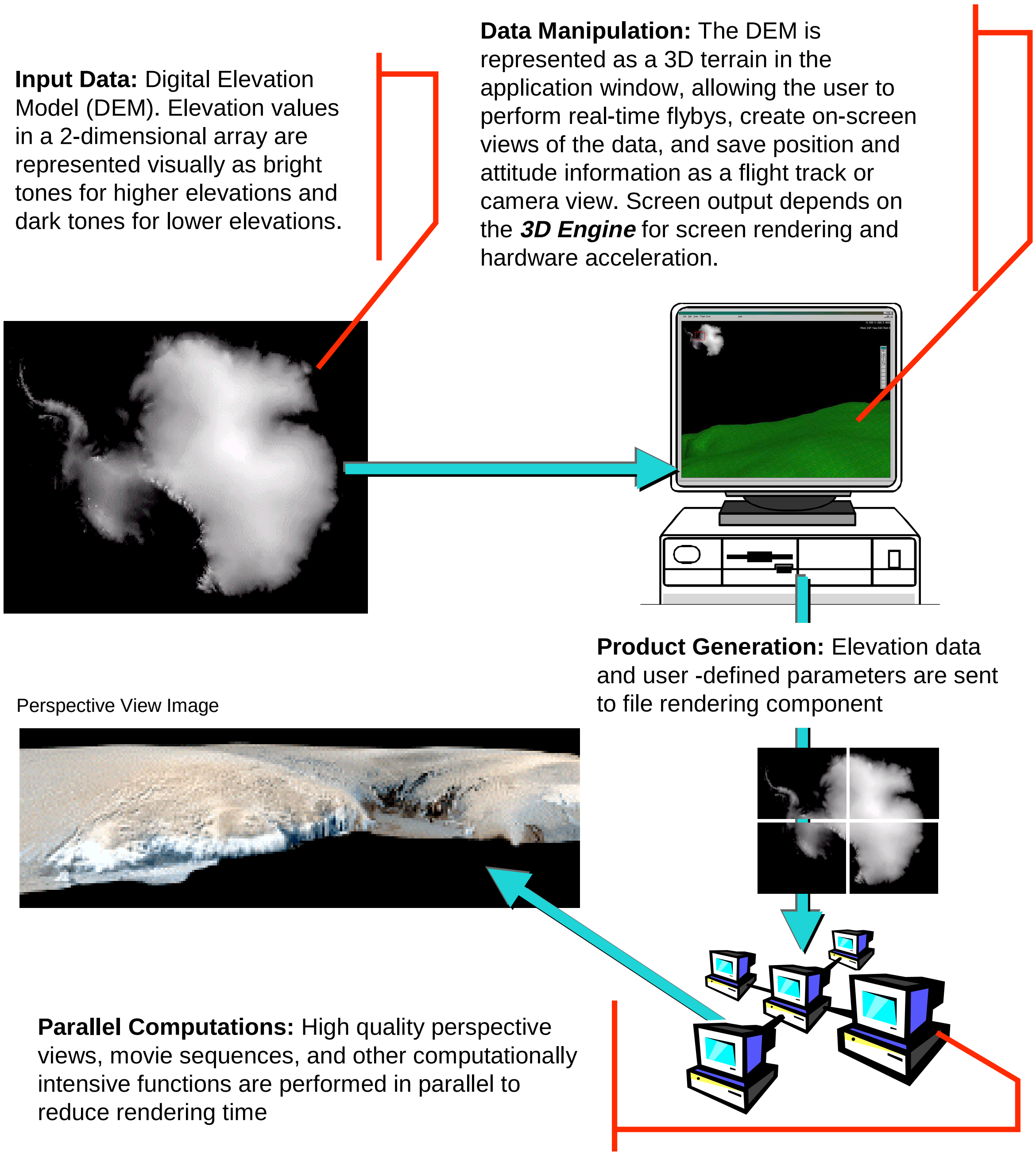


Figure 3 : Interaction of Components in the Terrain Modeling Package

Software Design

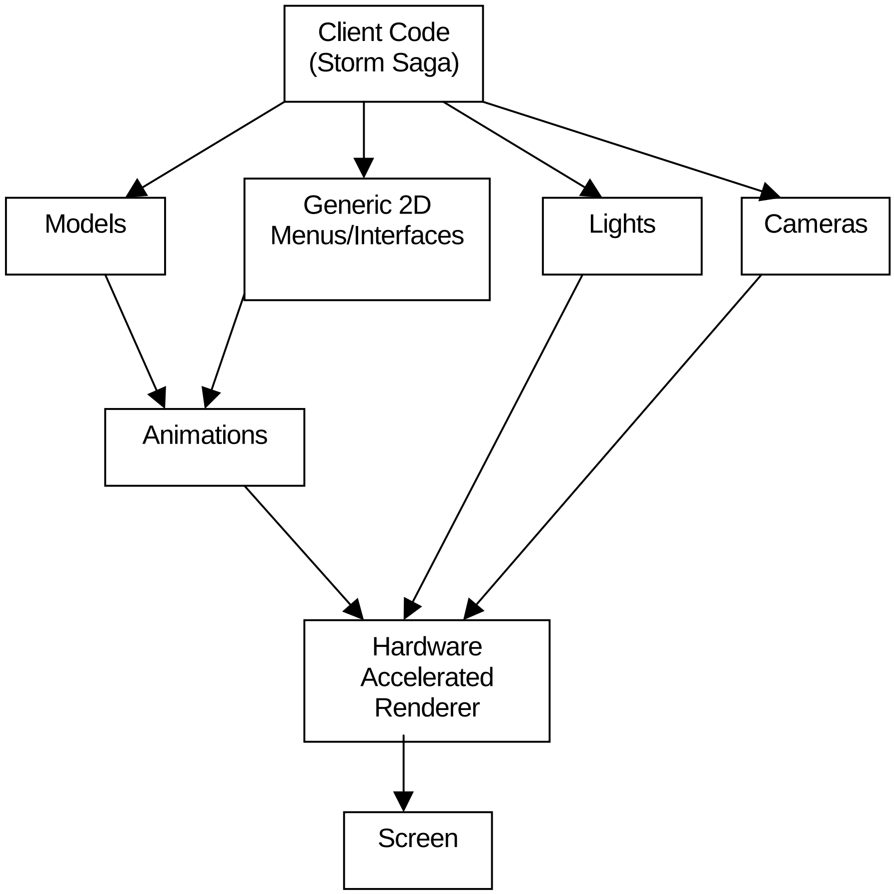


Figure 4:: 3D Engine Object Design

Terrain Modeling Package

Basic Architecture

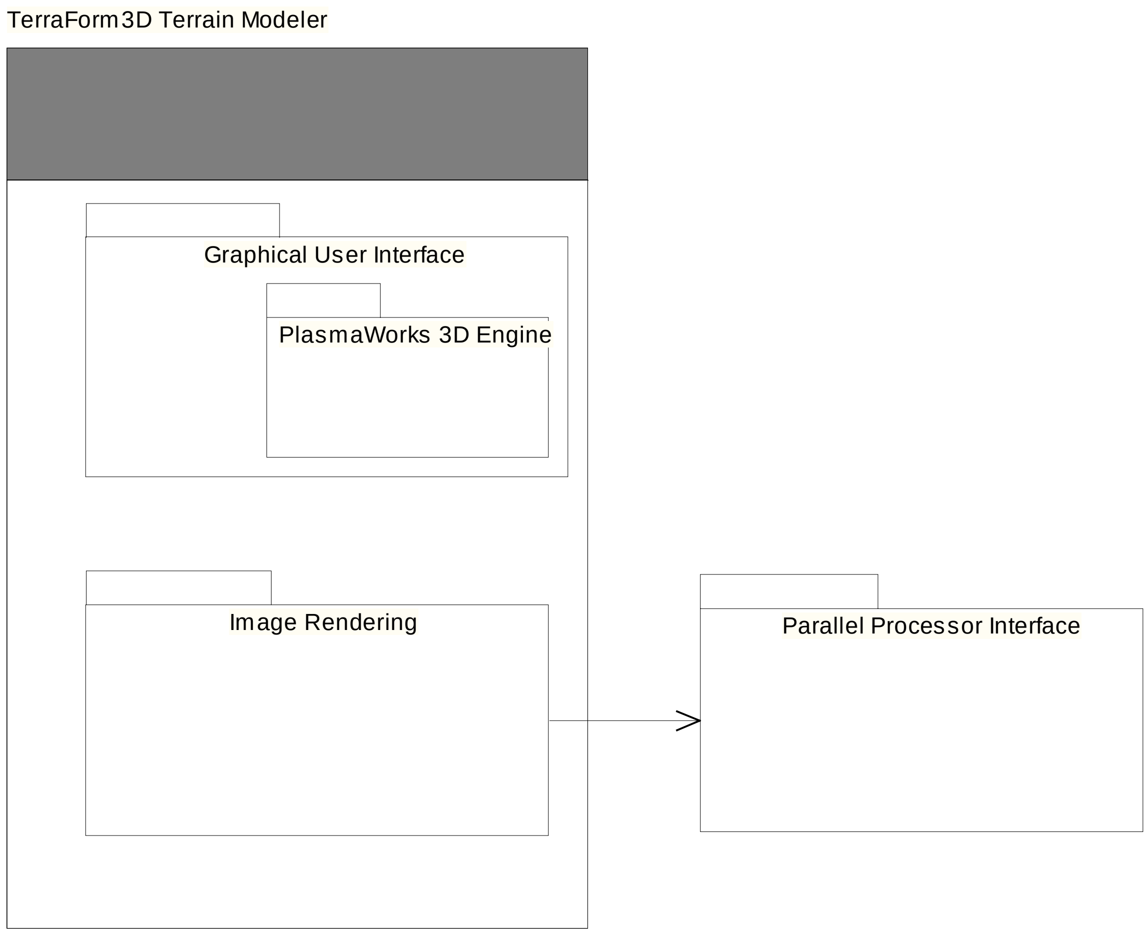


Figure 5 : Package diagram for TerraForm 3D System

User Interface

The figure below shows a prospective user interface with four main menus for accessing all functionality, a toolbar for accessing navigation and positioning functions, and a region showing the current position and attitude of the camera. In the upper left-hand corner is an overhead view of the loaded terrain with the location of the current viewpoint highlighted.

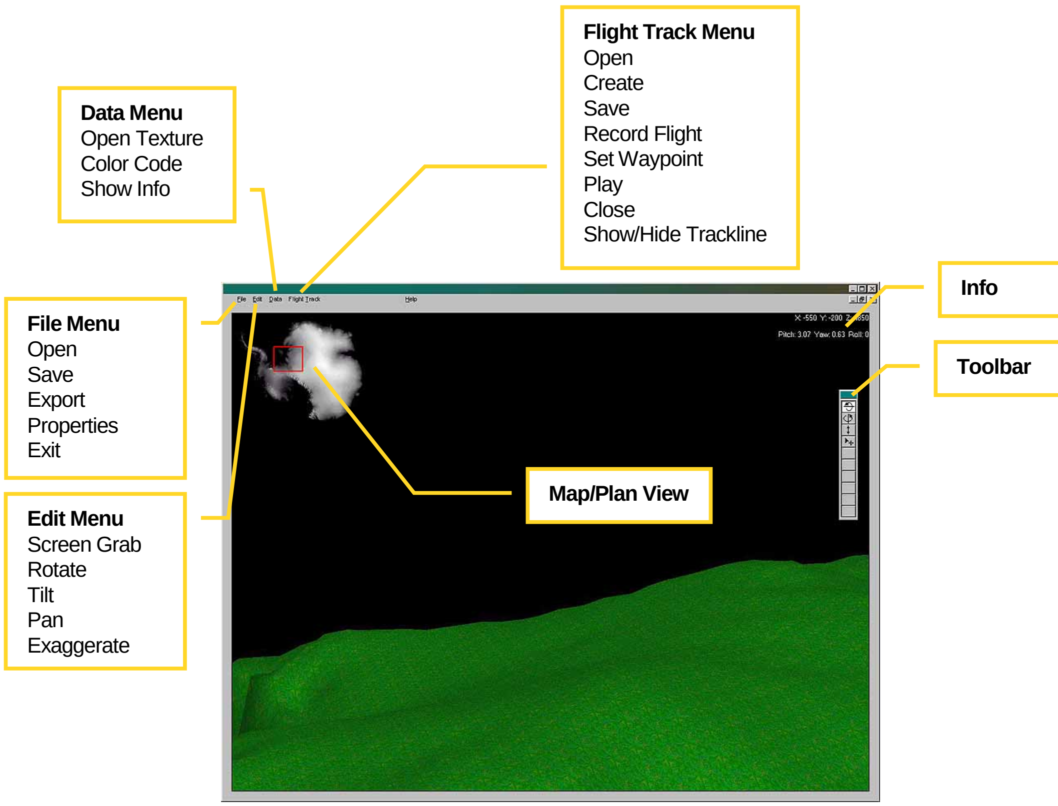
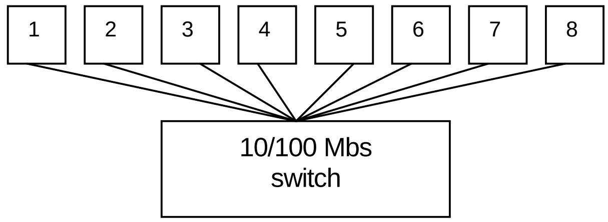


Figure 6 : Prototype user interface screen shot for TerraForm 3D System

Parallelization

Clustered System Design



Hardware	Quantity
Intel Pentium™ based systems	8
Memory	48 Megs per computer
Hard Drive	1 Gig per computer
10/100 PCI network cards	8
10/100 Mbs Ethernet switch	1

Figure 7 : Clustered System Diagram

The Linux based Beowulf cluster will be controlled through computer one. The master computer will be the only computer with a monitor and keyboard. All other computers will only be accessed via the master computer. The computers will be linked using Ethernet and a 10/100 Mbs switch, which is a device that allows all connected computers to transfer data simultaneously at full band width. This will keep each computer only one step away from one another. This combined with running a switch at 100 Mbs will give a performance increase.

User Interface

The master computer, using various LAM commands, will control the cluster. There will also be a monitoring program on computer one to observe the performance of the computers in the cluster. Authorized personnel will also be able to access the cluster from outside the USGS to run parallel programs.

Notes

Credits

Front Cover: *Artistically manipulated Antarctica terrain using screen output from DirectX prototype by Craig Post*

EndNotes

[1] Simulation of free flight over a terrain for the purpose of visualizing the terrain. Does not simulate aircraft, weather, or other factors common to flight simulation packages
[2] Graphical interface that allows a user to access a remote UNIX system and display software graphical user interfaces on the client machine
[3] Windows is Microsoft's operating system (O/S) for Intel-based systems
[4] Generates perspective views of a terrain
[5] Generates a shaded relief of a terrain
[6] VRML (Virtual Reality Modeling Language) is an ASCII file type representation of a 3-dimensional model which can be viewed and manipulated interactively, and is a popular distribution format for World Wide Web documentation
[7] Beowulf cluster is any number of mass-market off the shelf PC's interconnected by a local area network, running Linux, and executing parallel applications using an industry standard message passing model and library.
[8] Satish Reddy, Web Study Guide for Vector Calculus, Department of Mathematics, Oregon State University, http://iq.orst.edu/mathsg/vcalc/vcalc.html, 1996
[10][10] Compaq's UNIX O/S distribution for Alpha-based systems
[11] Digital Equipment Corporation's (DEC) UNIX O/S distribution for Alpha-based systems. Compaq acquired DEC in 1998 and renamed Digital UNIX O/S to True64 O/S
[12] SuSE is one of several commercial distributors of the multiplatform Linux O/S
[13] Sun Microsystem's UNIX O/S for SPARC- and Intel-based systems
[14] Apple Computer's O/S for Macintosh-based systems
[15] Silicon Graphics, Inc.'s (SGI) UNIX O/S for MIPS-based systems
[16] Clustered system communications and management software
[17] Message Passing Interface: programming environment for multiprocessor/clustered systems
[18] Graphics Library Utility Kit;
[19] GLUT is preferred because it was designed specifically for OpenGL
[20] Firewall is a way of keeping computers only accessible to authorized people.
[21] GNU Gnu's Not Unix. A software that is permitted to be modified and redistributed, but no distributor will be allowed to restrict its further redistribution. http://www.gnu.ai.mit.edu
[22] LAM Local Area Mulitcomputer, http://www.MPI.nd.edu/lam
[23] MPI Message passing interface, http://www.MPI-forum.org