TerraForm*3D*

Problem Statement

Draft 11 October 1999

Trent D'Hooge
Heather Jeffcott
Craig Post
Deborah Lee Soltesz

Table of Contents

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, the terrain modeling capabilities lack an interactive interface and many of the features available in contemporary terrain modeling packages. MIPS has a fully developed X-Windows/Motif user interface for all its programs, but does not have 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 entertainment product, named Storm Saga, which requires a three-dimensional object rendering engine (*3D engine*). The project is in its mid-design phase and Plasma Works have been planning to license a third party 3D engine. However, the company is interested in the analysis of the value and feasibility of the system, and 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 spherical coordinate system allows for easier rotations, weighted vertices, and easier model interaction.

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. 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, 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.

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 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 Modelling	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 incremental results 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.

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)	3	4
2.	Full texturing and lighting effects	4	3
3.	Animation-able Models	4	1
4.	A free floating camera	2	5
5.	Polygon culling	2	3
6.	Direct3d compatibility	2	5
7.	Opengl compatibility	1	5
8.	Legal agreement for use and ownership of 3D Engine.	1	5
9.	A solid, stable architecture that is easily modifiable	5	3

Spherical and Cartesian Coordinate System:

The Engine must be capable of both a Spherical and Cartesian coordinate system for storing vertex information. It must be able to display the benefits and hindrances of each system by performing benchmarking tests on both. Finally, it must be able to switch between systems without significant changes to the code.

Full Texturing and Lighting Effects:

Hardware Accelerated Textured Polygons must be supported. These polygons also must be lit by an array of lights including, but not exclusive to, bulb lights, spotlights, directional lights, and point lights. Colored lighting must also be implemented.

Animation-able Models:

The 3D Engine must support 3D models capable of animation, through either a bone structure interacting with a mesh of vertices, or a object structure interacting through an object hierarchy (i.e. the leg object is connected to the thigh object…).

A Free Floating Camera:

The camera pointing into the 3D scene must be "free-floating". It must be able to traverse the 3D landscape and orient itself to any position in 3-Space.

Polygon Culling:

The Engine must support backface polygon culling. It must be able to determine which polygons need to be rendered to the screen based on the camera location. It is imperative that this is done as efficiently as possible.

Direct3D/OpenGL Compatibility:

In order to maintain operating system portability, the Engine must support both the Direct3D rendering pipeline and the OpenGL rendering pipeline for Hardware Acceleration.

Legal Agreement for use and ownership of 3D Engine:

A legal agreement on ownership and rights to the 3D engine must be reached between Plasma Works and USGS. The 3D engine itself is being developed for Plasma Works with components of it being used and added on to by USGS

A Solid Stable Architecture that is easily modifiable:

A solid architecture is essential to the 3D Engine. This engine will be used by Plasma Works programmers later, and therefore must be easy to interface with, use, and exploit. It will also be maintained and modified by USGS staff and therefore must be easily modifiable.

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

* *Beowulf cluster – 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.*

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 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, which would allow other people to access source code and make changes that would improve the software
- Recognition by other institutions

Questions from the Team

The Senior Design team has the following questions for the sponsors:

- Will the engine have to have a hot-swappable coordinate system (i.e. change systems on the fly)?
- Will the team receive aid on algorithms/code problems?
- What are the frame rates with respect to system specification?
- Are there specifications and requirements for documentation, code structure, and the interface?
- What are some of the desired demonstrations?
- Does the engine need to have a software rendering mode as well as the hardware rendering mode?
- Will there be anyone at the USGS that would assist in running the Beowulf cluster?
- Is there anyone at the USGS who has experience in parallelization?
- Will the team be given access to development platforms?

Risk Analysis

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

Risks	Risk Analysis
The Spherical coordinate 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 so as 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.	We shall have to concentrate our 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 parallize the problem due to lack of experience or the problem as presented cannot be parallelized	Talking to the team that developed MIPS, we should be able to find a suitable way to run parts of the program in parallel. Parallelization of image processing applications and 3-dimensional rendering is well documented, and should provide a valuable reference in the design and development.

Benchmarking

The following summary and reference list shows what products were researched in order to learn what are the current state of research, development, and available consumer products, and potential developers' resources and references are available to the team.

Terrain Modeling and Scientific Visualization

Summary

The following table summarizes features found in popular terrain modeling packages:

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

Links to additional stand-alone and web-based software visualization examples
http://TerraWeb.wr.usgs.gov/TRS/resource.html

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

3D Engines, Rendering, and Applications

Summary

Current 3D Engines on the market are capable of a wealth of features. All support some level of texturing, lighting, animation, and polygon culling. They also come with tools to aid the designers in use of the 3D Engine. These tools include: 3D Modelers, Animators, Level Designers, and Texture Mappers.

The more competitive 3D Engines offer advanced visual effects such as lens flare, fog, fire effects, weighted vertices, and deformable objects. There are three engines of specific note on the market today, the QuakeIII Engine, the Unreal Engine, and the Lithtech Engine. There are also numerous smaller less well known (and less expensive) engines available to the consumer market.

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/

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/

Various papers, examples, and explorations on rendering, modeling, geometry, and other computing and 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

Parallelization

Government and educational institutions are currently implementing most clustered systems. Beowulf clusters are commonly used for applications like security and image processing. Because of the difference in problems being solved or proprietary software created for the systems, the software cannot be sold or given to other institutions to be reused. There are common threads between all Beowulf clusters though, such as using message passing like PVM or MPI. They also all run a Unix-like, open-source operating system, such as Linux. Most implementations of Beowulf clusters are also well documented by the various institutions, and the vast resources of help on the Internet for Beowulf clusters makes using this type of implementation for parallization a good one.

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

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/

Other Resources and References

Open Source Remote Sensing http://www.remotesensing.org/