Serious Games for 3D Seismic Travel Time Tomography

Ivan Gris
Faculty Mentor: Rodrigo Romero
Cyber-ShARE Center of Excellence
University of Texas at El Paso
El Paso, TX, 79968, USA
Email: igris@miners.utep.edu, raromero2@utep.edu

Abstract

Serious games and game technologies are commonly applied for training, education, and simulation environments. In this paper, we present an approach that uses game development technologies to enable analysis of Earth crust velocity models generated with seismic travel time tomography. We created a training application that was implemented as a serious game with a 3D interface facilitating exploration of tomographic processing outputs. The application features color-keyed extrapolation maps of tomographic experiments and real-time interactive visualizations that provide students and researchers an environment to explore and learn about the details of computed three-dimensional velocity models and related tomographic models. Utilizing an open source game engine within a three-layer architecture, the result is a serious game environment that provides a detailed model of a section of the crust of the Earth through a three-dimensional extrapolated map.

1. Introduction

As applications for computer simulation have gained acceptance in many commercial, industrial, and military areas, new applications have emerged. These applications have been used for "serious" training purposes as well as for education and entertainment [1]. Many types of serious games have been implemented for educational purposes in almost every area that requires training. Using games to learn is a new experience that has proven to be effective in businesses and schools. Although games have been created to teach within several scientific areas, serious games are often aimed at high school students or at proven experiments where the final solutions are known.

This paper presents a three-layer architecture for creating experimentally driven 3D serious games. The produced game will assist geologists to create more accurate models of the crustal velocity structure of the Earth. In addition, the game will help geology students to understand the details of a travel-time seismic tomography algorithm and to create accurate velocity models in real time, while engaging in motivating, learning experiences. Interactive three-dimensional visualization facilitates the interpretation of tomography results beyond the possibilities offered by standard orthogonal projections found in hard-copy renderings of models.

2. Game Engines

A computer game is a goal-directed and competitive activity that may involve some form of conflict [2]. A game induces independent decision making in users, as they must seek a successful approach to achieve objectives within the game context [3]. Games fall into a wide spectrum of genres, which are often combined to create games with the best elements of two of more genres. Thus, a game of the serious game genre can be enhanced with elements of the strategy and role-playing genres to create a well-rounded playing and learning experience for users, who are referred to as players. A distinguishing characteristic of serious games is that they have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement [3].

Good software design practice dictates that games, which have complex calculation requirements, an extensive array of features, or support for multiple, diverse input devices, be architected with a lower layer of software known as a game engine. A game engine abstracts the implementation details of tasks, such as rendering, physics modeling, and low-level input/output processing. Use of third-party game engines allows game developers (graphic designers, musicians, story writers, level designers, scripters, and programmers) to focus their efforts on the details that make their games unique [4][5].

Game engines also facilitate rapid prototyping of conceptual ideas with computationally intensive

simulation environments and complex game dynamics. Typical engines include reusable components to implement model handling and display, collision detection, physics, multi-device input, graphical user interface, and artificial intelligence. In contrast, the components that comprise the content of the actual game include the meaning behind object collisions, the responses to player input, and the way objects interact with the world within each level are [4]. An ancillary type of game engine, referred to as a middleware engine, specializes on specific functions and works in conjunction with the main game engine. While keeping within the frame budget, middleware engines perform computationally intensive tasks with a higher degree of realism and a more varied functionality than a main engine. Middleware engines typically implement physics, artificial intelligence, path finding, cloth simulation, destruction simulation, and user interfaces [6].

3. Seismic Travel Time Tomography

Seismic travel time tomography is a technique used to investigate the velocity structure of the Earth's crust. The algorithm presented in this paper is an iterative application of forward modeling and non-linear inversion designed and implemented by J. E. Vidale [7] and J. A. Hole [8], respectively. Forward modeling uses the eikonal equation to compute a discrete three-dimensional first arrival time model for each seismic wave source included in a tomographic experiment. The inversion procedure uses the forward modeling output to compute the source-to receiver wave propagation path and the first arrival time for each source and receiver pair of an experiment. The algorithm concludes when experimentally measured first arrival times and propagation times calculated during inversion agree within rms values of measurement error. The final outputs of the tomographic algorithm include a three-dimensional velocity model and seismic ray coverage of the model.

4. Game Architecture

The main goals of a serious game for seismic tomography are three-dimensional display and manipulation of velocity models and associated tomographic models, interactive visualization of intermediate and final models, and a minimalist feature set to enable system operation without a prolonged learning curve. These game-inspired goals contrast with the lacking visualization capabilities, the text-based semi-batch mode of execution, and the lengthy running time of the underlying tomographic software, which as a whole discourage user experimentation with incremental model validation and evaluation of what-if scenarios.

The system design is based on a three-layered architecture illustrated in Figure 1. The game engine implements the bottom layer. The middle layer, where most games use a middleware engine, implements the Vidale-Hole seismic tomography algorithm. The top layer implements visualization and model manipulation control. The player's main goal is to adjust processing parameters until the seismic tomography algorithm converges to a valid velocity model, while subgoals include qualitative correctness of intermediate results and a general trend of intermediate results toward velocity model convergence.

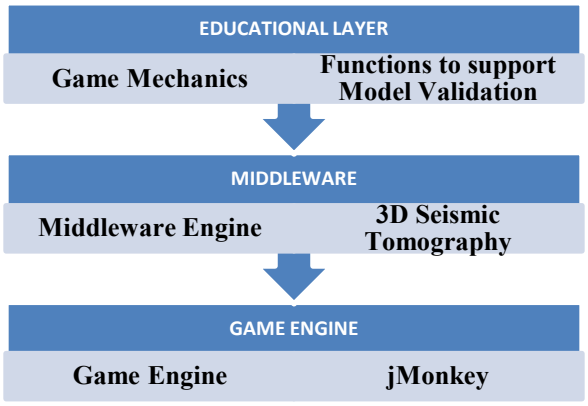

Figure 1. Three-layered architecture of the serious game for seismic tomography

5. Game Implementation

The bottom and the middle layers of the game are fully functional third-party building blocks, which are advantages of using a game engine and an existing implementation of a seismic tomography algorithm. Thus, most of the implementation effort of the serious game was dedicated to the top layer to meet game design and implementation goals.

Depending on model size, the game offers interactivity and frame generation rates in the order of one frame per second, which compares favorably with the alternative of manual display of the final

velocity model after approximately an hour of processing. In addition, visualizations are color-keyed to indicate model details such as cell hit count when rendering coverage, as shown in Figure 4, which compares favorably with respect to the usual manually generated fixed 2D orthographic projection, gray-scale rendering of the same information. The serious game implemented creates a rich 3D environment that is fully navigable in real time to support and encourage interactive model exploration. The game also implements a spatial extrapolation in several color schemes that provide a simple interface for displaying ray concentration per cubic kilometer.

The game was written in Java using jMonkey as the game engine to benefit from portability at the source code, CPU architecture, and OS/GUI levels [9]. jMonkey also provides a camera implementation, graphics primitives, networking capabilities, and an interactive frame generation performance.

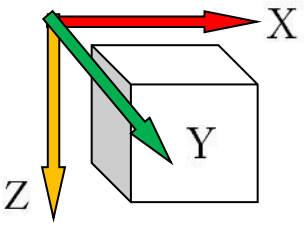

Figure 2. Representation of 3D space with arrows pointing to the positive direction.

The top layer controls creation of the tomographic model 3D grids by traversing the model in slices and associating tomographic output information for each cubic kilometer. The seismic tomography algorithm outputs one file per model which contains binary values of model vertex or cell information according to each model. Output models represent crustal velocity information in a left-handed coordinate system, but the positive Z direction, which corresponds to depth, actually points up in model coordinates. Visualizations are based on a set of display transformations that result in crustal depth, the Z axis, as shown in Figure 2, correctly increasing downward and starting from zero at the surface; horizontal measurement along the X-axis increasing from left to right; and horizontal measurement along the Y axis increasing from front to back. Model X axis or Y axis orientation with respect to the Earth north is determined by the user and is independent from the tomographic experiment layout.

Using generation of model coverage visualization as an example, which is shown in Figure 3, every time a primitive is aggregated to represent a model cell, a red, green, or blue value is assigned to the primitive according to the number of seismic wave rays passing through it. Where there is a zero in the model file, no rays travel through the cell and no primitive is added to the visualization to avoid cluttering coverage display. The game takes a coverage binary file, translates cell hit values to integer form, and creates a primitive at the cell position with a color that maps to the cell hit count. After the entire model is loaded, the user gets control of the camera to explore the model in real time.

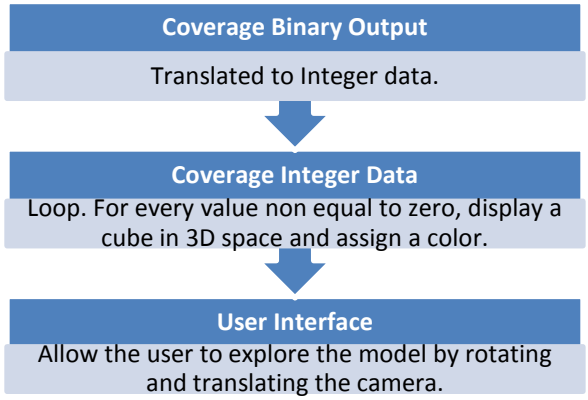

Figure 3. Implementation steps to visualize 3D model coverage.

6. Conclusions

This paper describes the design and implementation of a serious game to visualize, analyze, and control the outputs of an implementation of first-arrival travel time seismic tomography. The game offers interactive, color-keyed, three-dimensional visualizations of seismic tomography outputs to encourage students and researchers to learn about and analyze the details of produced three-dimensional velocity models and related tomographic models.

The game enables model visualizations as output is generated, which is in contrast with batch computations and subsequent manual visualization generation associated with standard use of the underlying seismic tomography software. Since the tomography software runs several iterations and may

take a few hours to complete for complex models, real-time visualization of iteration outputs will speed up model exploration for qualitative validation and convergence assessment.

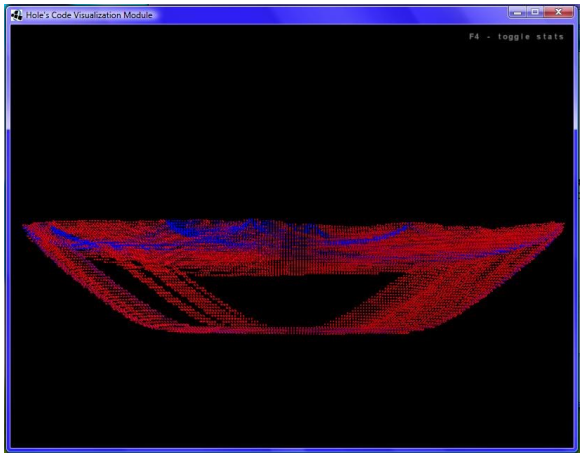

Figure 4. A sample ray coverage output.

Future work may enhance the game with the addition of visual clues of lack of model convergence, parallelization of the seismic tomography software, and selective control of visualized models.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation under Grants CREST HRD–0734825 and CNS–0837556.

References

[1] V. Narayanasamy, K. W. Wong, C. C. Fung, and S. Rai, "Distinguishing Games and Simulation Games from Simulators," ACM Computers in Entertainment, Vol. 4, No. 2, April 2006

[2] L. Sauve, L. Renaud, and D. Kaufman, "Games and Simulations – Theoretical Underpinnings," Proceedings of the Digital Games Research Association Conference, Vancouver, B.C., 2005

[3] C. Abt, "Serious Games," 1970

[4] J. Ward, "What is a Game Engine," http://www.gamecareerguide.com, as seen in 2008

[5] R.E. Pedersen, "Game Design Foundations" 2nd Edition, 2009

[6] "Havok AI, Cloth and Physics" Havok.com Inc, 2009

[7] J. E. Vidale, "Finite-Difference Calculation of Travel Times in Three Dimensions," Geophysics, 55, 1990

[8] J. A. Hole, "Nonlinear high-resolution three-dimensional seismic travel time tomography, J. Geophys. Res., 97, 1992

[9] M. Roulo, "Java's Three Types of Portability" http://www.javaworld.com, 1997