Visual Analysis of Seismic Simulation Data

Florian Juergen Gerhardt*
Technical University of Kaiserslautern, Germany

Joerg Meyer†
Electrical Engineering and Computer Science Department
University of California, Irvine

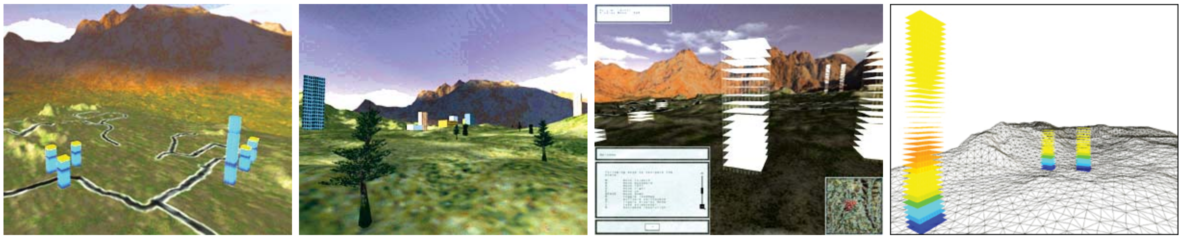

Figure 1: (a) Symbolic representation of dynamic building response to seismic activity superimposed on roadmap. (b) Photorealistic display mode, including a set of trees and a skybox. (c) Semi-expert display mode and GUI (before earthquake, neutral building color). Bottom right shows a bird-eye view of the scene and the current user's position for orientation. (d) Scientific display mode, including seismic activity (magnitude of displacement mapped to color of building floors).

ABSTRACT

Seismic simulations use finite element methods to describe ground motion. The results of such numerical simulations are often difficult to interpret for decision makers. We describe a terrain rendering engine that uses photorealistic metaphors to represent typical terrain properties without representing an actual terrain. In the context of ground motion, a simulation of the effects of various types of earthquakes on buildings has been conducted. Usually, such structural response simulations are carried out independently and are being visualized separate from the ground motion simulation. We combine the results from both simulations in an interactive, hybrid visualization so that decision makers (first responders and emergency management agencies) are provided with a photorealistic, simulated view of various earthquake scenarios, enabling them to study the effect of various earthquakes on buildings typical for a rural or urban area.

We present a method for visually analyzing large-scale simulation data from different sources (ground motion simulation and structural response simulation) using photorealistic metaphors. We have implemented an intuitive, interactive system for visual analysis and inspection of possible effects of various types of earthquakes on an inventory of buildings typical for a particular area.

The underlying rendering system can be easily adapted for other simulations, such as smoke plumes or biohazards.

Index Terms: I.3.3 [Computing Methodologies]: Computer Graphics—Picture/Image Generation; I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism

1 INTRODUCTION

Seismic finite element simulations produce large, time-varying tetrahedral meshes representing ground motion during earthquakes. Of specific interest to Civil Engineers are the surface effects of earthquakes on the various types of buildings typically found in an urban setting. The surface is represented as a triangle mesh, which is derived from the tetrahedral mesh [1]. In the past, ground motion simulations and building simulations were usually decoupled, because the underlying models are different. While ground motion simulations require tetrahedral meshes to represent elasticity and dampening effects of soil, building simulations are usually based on structural models (stick and mass) which are excited at the bottom of the structure. A time-varying, 3-D displacement vector, which controls the excitation, is used to model various types of earthquakes. This vector can be generated by a simple, attenuated, oscillating function, or it can be derived from recorded data from real earthquake events.

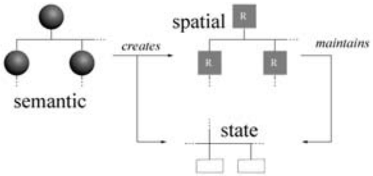

Figure 2: Scenegraph context

In our system, the displacement vector is derived from a point on the surface of a large tetrahedral mesh, which is used for ground motion simulation. This mesh is a time-varying, 3-D representation of a soil model. While the mesh nodes are not displaced, each node contains a maximum velocity vector, from which a displacement vector can be derived, which indicates direction, orientation and magnitude of shockwaves traveling through the node. Since we are only interested in the surface effects of an earthquake, we extract the surface layer triangle mesh from the underlying tetrahedral mesh and map it onto a terrain model. The terrain model consists of a height field and a texture. The texture can be either color-coded elevation data or, for instance, geo-referenced map data.

For the buildings, the underlying stick and mass model is rendered in the form of a frame structure, where four columns move in parallel (sticks), and each floor is represented by a concrete slab (masses) rendered as a quadrilateral.

The novel aspect of our work is a visualization which is inspired by tourist maps which feature exaggerated representations of landmark buildings on 2-D terrain displays. Our buildings represent

---

[*] e-mail: gerhardt@mv.uni-kl.de
[†] e-mail: jmeyer@uci.edu

buildings that are typical for a particular neighborhood (high-rises, mid-sized office buildings, or primarily residential areas). We have developed this idea further and created a time-varying, 3-D visualization driven by simulation data, which is used for visual analysis of simulation data from different sources. A key component of our system is its ability to facilitate changes to the map, the earthquake type, or the building inventory and to visualize the altered scene immediately for visual inspection and analysis.

The goal is not to predict earthquakes, which is currently still impossible, but to simulate different scenarios, so that first responders and emergency management agencies can be better prepared in the event of a natural disaster.

2 DYNAMIC RENDERING FOR VISUAL ANALYSIS

The established concept of a scenegraph is used to represent and organize complex scenes. In our approach, we separate the semantics from the spatial and current state information and store each type of information in a separate subtree. The following sections explain the purpose and structure of each sub-scenegraph (figure 2).

2.1 The Semantic Scenegraph

The semantic scenegraph defines the logical view of the visual database. It features a set of nodes that objects can be derived from. Group nodes usually possess an arbitrary number of children. It is this type of node that stores unique transformation information, meaning a translation vector, and a rotation quaternion. Upon the scenegraph's update traversal, this information gets pushed onto a history stack, until a leaf node is hit.

A basic leaf node is derived from various leaf types, defining properties of the represented objects. If an object, for example, is renderable and deformable, the renderable leaf interface and the deformable leaf interface can be derived from it.

A so called *decorator node* is a group node that is only allowed a single child. It is often used to *decorate* a leaf node's transformation. Figure 3 shows an example of a semantic situation.

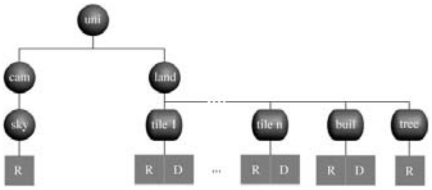

Figure 3: Semantic scenegraph example

■ depicts a leaf node, R = Renderable, D = Deformable

● depicts a group node

▬ depicts a decorator node

2.2 The Spatial Scenegraph

The spatial scenegraph is automatically generated from the semantic scenegraph, solely using the renderable nodes existent in the current scene. It defines the bounding volume hierarchy of the current scene.

Each spatial node encloses the contents of the branch below. For an arbitrary point in time, each node is checked for visibility. Should the user's eyes not see the node, then all nodes directly or indirectly attached below in the tree are also not visible and do not need to be rendered. This hierarchical structure has a significant impact on rendering times for complex scenes.

A bounding volume update follows the semantic scenegraph's general update traversal. Recalculation is done bottom up, i.e., starting from the leaf nodes moving up towards the root. A renderable's bounding volume is defined by its vertices. This volume is propagated higher and merged with other volumes, so that each group node represents the joint bounding volume of a children nodes.

With regularly updated bounding volumes, the existing spatial scenegraph manages visibility of objects. In each frame of display, the modelview matrix and perspective matrix extract a *viewing frustum*, defining the visible volume of space.

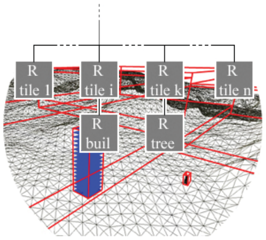

Figure 4: Spatial scenegraph example

The spatial scenegraph is traversed in order to test each object's spatial boundaries as a whole against the frustum. If it finds the object to be outside the pyramid, the object is culled away, and its polygons are no longer considered. Furthermore, all of the node's children objects must also be located outside the frustum. Hence, on its spatial branch, no further testing is needed in the traversal.

2.3 The State Scenegraph

The application interface maintains an automatically generated state scenegraph, consisting of all faces in the active scene, sorted by applied materials. The effect is a reduction of GPU overhead due to fewer state changes during the rendering traversal.

All faces sharing the same material are to be found on the same branch. The tree depicts which faces are currently visible. Visibility information is reset for each displayed frame and updated upon traversal of the spatial scenegraph. The spatial scenegraph is traversed and the state scenegraph is updated.

3 CONCLUSIONS

An interactive seismic application was created on top of a scenegraph-based visualization framework for visual analysis of time-varying deformation data of terrains and buildings. A scenegraph, which can be updated through the user interface for different maps, earthquake types and building inventories, was used to model semantic, spatial and state information. Despite the fact that many data structures need to be updated, the tree structure proofed to be efficient, and interactive frame rates between 8 and 44 fps were achieved on commodity PC hardware (Windows Vista Ultimate, 3.1 GHz CPU, ATI Radeon X800 Graphics Card).

This work was supported by the National Science Foundation under contract 6066047–0121989 (SPUR - Seismic Performance for Urban Regions). We would like to acknowledge Gregory L. Fenves and Bozidar Stojadinovic, Department of Civil and Environmental Engineering, University of California at Berkeley (structural response simulation), and Jacobo Bielak and Antonio Fernández, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, MA (ground motion simulation).

REFERENCES

[1] Joerg Meyer and Thomas Wischgoll. Earthquake visualization using large-scale ground motion and structural response simulations. In *Scientific Visualization: Extracting Information and Knowledge from Scientific Data Sets*, pages 409–432. Springer, 2005.