THURST AREA #2:
Virtual & Synthetic Environments

TITLE: ARC 2.1
Off-road Synthetic Environment Modeling & Simulation

Presenter: Yiannis E. Papelis, Ph.D.
Research Scientist - NADS & Simulation Center
Research Participants

- **Tire-soil interaction modeling**
  - Dr. R. Ranganathan. (RS, NADS)
  - Dr. Saib Othman (RS, NADS)

- **Terrain Database Modeling**
  - Dr. Yiannis Papelis (RS, NADS)
  - Imran Pirwani (graduate std., CS)

- **Dynamic Terrain Visualization**
  - Dr. Jim Cremer (Fac., CS)
  - Yefei He (graduate std, CS)
Quad Members Participation

- TARDEC
- Computer Science Dept., U of Iowa
- John Deere
- NADS & Simulation Center
Talk Outline

- Define Goals for this task
- Provide brief description of existing work to build upon
- Overview of tire-soil interaction models
- Cover work on dynamic terrain database models
- Results on visualization techniques
- Provide current status
- Define goals for the next 6 months
Primary Goals

- Physics-based modeling and visualization of the vehicle-terrain interaction
  - Investigate Tire-Soil dynamics
  - Refine static terrain models with dynamic terrain capabilities
  - Investigate or derive dynamic terrain visualization capabilities
System Framework

- Synthetic Environment Visualization
- Tire-Soil Simulation
- Multi-Resolution Dynamic Terrain Representation
- Visual Database (E&S GDF, or OpenFlight)
- Resolution & Attribute Specification
- Database interface
  - QueryGrid()
  - ModifyGrid()
- Terrain Database Using Variable Res Grid
- Scanner
- Compiler
Real Time Tire-Soil Interaction Modeling

- Bekker’s Soil Model Formulation
- Integration of Tire/Soil Interaction Model with Vehicle Model
- Couple with the Terrain and Visual Database
Bekker’s Soil Model Formulation

- Generate Position and Velocity of the Wheel Center
- Query for Terrain Elevation/Properties
- Position the Wheel in the Soil Coordinate System
- Obtain Soil Deformation $Z_r$ Using Bekker’s Formulation - Iterative Search
- Compute Soil Reaction Forces
Modify Algorithm To Reduce Computational Time

- Secant Search Algorithm Using Smart Initial Estimates; Search Converges in 1 to 2 Iterations (Computational Speed Increased Four Times)
- Modify Integration Algorithm to Integrate Pressure and Shear Force Functions at the Tire/Road Interface Simultaneously (Two Times Faster)
- Use of Pre-Computed Carpet Plot for Lateral Force Computation on Deformable Terrain
Existing Work in Terrain Databases: Models

• General Requirements
  – Arbitrary resolution
  – Realtime performance at high query rates (>20kHz)
  – Support for large volumes of data (larger than memory)
  – Support special effects (water bodies, surf. Properties)
  – Support overlapping terrain (bridges, tunnels etc.)
  – Compatibility with existing visual databases
Existing Work in Terrain Databases: Basic Model

- Use uniform gritted terrain within rectangles (Datazones)
- Terrain database comprised of multiple overlapping datazones
- Bilinear interpolation within quad
Existing Work in Terrain Databases: Interrogation Algorithm

- Find datazone(s). For each do:
- Find 4 surrounding elevation posts
- Use bilinear interpolation., find z and normal
- Pick best candidate based on input Z
Existing Work in Terrain Database: Dynamic Paging

- Small *working set* of data is needed on-line
- A look-ahead loader reads data ahead of time
  - Working set stored in memory
  - Unlimited database size
  - Throughput of data depends on velocity, data density

```
| Terrain Database Using Variable Res Grid |
| Lookahead Loader |
| Working Set |
| x, y, vel |
| z normal vec |
| Dynamics Model |
| Tire-Soil Simulation |
| Interrogation Code |
```
Existing Work in Terrain Database: Database Creation

• Sources:
  – E&S visual databases
  – OpenFlight databases

• Tools
  – Scanner: read a datazone layout, generate datazone grid
  – Compiler: package datazones, build data structures, output final terrain database
Dynamic Terrain Enhancements
ARC work

- Terrain Database Using Variable Res Grid
- Working Set
- Lookahead Loader
- Dynamics Model
- Tire-Soil Simulation
- Interrogation Code
- x, y, vel
- z normal vec
Dynamic Terrain Enhancements

- All data flows from/to the terrain database become bi-directional

- Data throughput analysis for terrain:
  - \( T_d = \left( 2\times W \times v / \text{res}^2 \right) \times \text{DSS} \)
  - \( W \) Length of line
  - \( v \) Velocity of vehicle
  - \( \text{res} \) Resolution of terrain
  - \( \text{DSS} \) size of a dataset

- Example:
  - 100 mph, 44 ft line, 2 in., 12 bytes
  - Data rate \( \sim \) 6.5 Mb/Sec
Interface with Tire-Soil Module

- Tire-Soil module expects a uniform grid aligned to the tire longitudinal axis
- Input to Tire-Soil:
  - Interrogate on the tire grid
- Output from Tire-Soil:
  - Solve reverse bilinear equations to determine effect of elevation changes from tire grid to original grid
Applying elevation changes

- On input to Tire-Soil, point p is evaluated using elevation posts p1…p4
- The result of Tire-Soil interaction provides a point p with a different Z and different location
- Must find effect of new p on p1…p4
Computing Effect on Original Grid

- Some of the problems
  - Multiple solutions for effect of $p$ on $p_1 \ldots p_4$
  - Effects of adjacent quads on elevation posts add constraints
  - Real-time execution when sweeping tire grid
  - Coarse resolution or original grid may not be able to capture high resolution effects
Dynamic Terrain Visualization

- Lots of recent work in multi-resolution (MR) or multi-LOD object and terrain representation
- Most methods assume static data, allowing use of substantial preprocessing
- For visualization, of interest is the ability to view multi-resolution data from different viewpoints
  - appropriate LOD levels determined by error criteria
  - extremely attractive are error terms described in screen units (pixels)
Dynamic Terrain Visualization
Goals and Activities

- Develop or adapt an MR algorithm for dynamic terrain visualization
- Focus on existing algorithms with desirable constraints (pixels)
- Currently, existing algorithm (depending on static data) has extended for dynamic data
  - Implemented w/ Performer (SGI), OpenGVS (PC)
  - Good performance for small databases, but computationally expensive for larger
Dynamic Terrain Visualization
Approaches to Performance

• Option 1: use single MR dynamic database representation
  – e.g., use visualization algorithm over entire database

• Option 2: use a hybrid MR dynamic database representation with traditional LOD
  – e.g., use standard LOD techniques with static terrain when far, switch to MR adaptive technique when near
Hybrid LOD-MR Adaptive Technique

static LOD

adaptive MR
Current Work

- Determine efficient solutions for the bilinear equations, including adjacent quads
- Benchmark performance
  - Initial results are promising: work can be pipelined
- Investigating dynamic generation of new data zones during sweeping
Status

- Most components are in isolation
  - Tire-soil, terrain database, MR algorithm
- Goals:
  - integration of subsystems
  - evaluation of real-time performance
  - refinement and optimizations