DEM Compression and Terrain Approximation; Smugglers and Border Guards

Prof W Randolph Franklin, Prof Frank Luk, Prof Barbara Cutler, Prof Marcus Andrade, Metin Inanc, Zhongyi Xie, Dan Tracy, Jon Muckell
Rensselaer Polytechnic Institute
mail@wrf franklin.org, 703-447-7808

Complete Record

- This version contains both the slides that I showed to NGA (on 10/1/2007), and to DARPA (on 10/2/2007). Therefore it is both a good introductory and detailed record of RPI’s performance to date.
- [http://www.ecse.rpi.edu/~wrf/pmwiki/GeoStar](http://www.ecse.rpi.edu/~wrf/pmwiki/GeoStar) contains most material ever given by RPI to DARPA or NGA.

Ask WRF for a password
Team

- Prof Randolph Franklin – helping everyone
- Prof Barbara Cutler – computer graphics
- Prof Frank Luk (on leave as Vice-President (Academic) of Hong Kong Baptist U) – numerical analysis
- Prof Marcus Andrade – visiting from UF Viçosa (Brazil) – computational geometry
- Metin Inanc – ODETLAP
- Zhongyi Xie – ODETLAP
- Dan Tracy – multiobserver siting, path planning
- Jon Muckell – hydrology.

Potential Benefits

- More compact terrain representations.
  - Store the ever larger amounts of terrain data. Spend time on the compression (which is done once) to get most compact representation.
  - Works on 16 bit topography.

- Conflate overlapping inconsistent cells.
  - Overlay large, low precision cell by smaller high precision cells => one unified elevation field.

- Efficient routing by nonspecialists.
  - Route our aircraft away from antiaircraft batteries

- Better hydrographic computations.
  - Flood prediction.
An Inadequate Terrain Representation
Goal 1: DEM Compression and Terrain Approximation

- ODETLAP: alternate terrain representation.
- Compact.
- Allows lossy - size / quality tradeoffs.
- Emphasized decompression speed.
- Evaluated on visibility, mobility metrics.

Milestone Progress

- Phase I: 10x compression while maintaining usefulness; Phase II: 100x
- Reverse engineered HRTI Analysis Tool’s slope formula to avoid running HAT each time.
- We gave before-and-after data to NGA demonstrating this.
- Further improvements made since then.
- Fitting other components of proposal (e.g., hydrology) to correspond to milestones.
**Key Differentiating Factors**

- Smooth representation
  - no visible blockiness

- Allows progressive transmission
  - longer transmission => more accurate reconstruction

- Conflates inconsistent partially overlapping data.

- Interpolates partial sets of elevation posts
  - generates continuous slopes even when the input data consists of nested contour lines.

- Infers local maxima inside the topmost contour.

---

**ODETLAP Process**

Since our first description of ODETLAP at the 1998 Spatial Data Handling Symposium, we’ve built this system.
**ODETLAP Point Selection**

- Incremental TIN to find most important points, then greedy insertion of worst points (*Allows progressive transmission*)
- Regular grid of points (more points, but compress better) *(More compact)* **NEW**
- Stream and ridgeline points *(Preliminary)* **NEW**

---

**Other Point Selection Strategies**

The following were not as good:

- Select highly visible points, *or*
- Random points, *or*
- Points based on histogram of heights with boosted sampling for less frequent elevation bands and small connected components.
Incremental Triangulated Irregular Network (TIN)

- Can process $10^8$ points on a laptop.
- Works in memory w/o needing to page data from disk.
- Inserts points incrementally, in order of importance.
- Can progressively transmit terrain.
- Identifies ridge lines automatically.

Coding the Points to Reduce Space

- Code $(x, y)$ separately from $z$. \textit{NEW}
- If $(x, y)$ a regular grid: give its resolution
- Else: run-length encode the bitmap.
  - 0100000011000010001 -> 16043
  - Only about 1000 of 160,000 bits are 1.
- $Z$: delta code, then \textit{bzip2}.
  - 100 125 90 90 100 -> 100 25 -35 0 10
Traveling Salesman Path

- **NEW!**
- Hypothesis: nearby points often have nearby Z, which delta code better
- Find a traveling salesman path through the selected points.
- Put the Z in that order and code them.
- (X,Y) coding is not affected.
- **Status:** have some preliminary results.

Info theoretic min for (x,y)

- Assume that 1000 of 400x400 bits are 1, rest are 0.
- Assuming no structure in the 1s, size is $lg(\text{choose}(160000,1000)) = 8754 \text{ bits}$
- We approach that within 20%.
- That’s why we separate (x,y) from (z).
Better Than the Info-Theor Limit?

- The information theoretic limit was calculated assuming no structure in the points.
- Is there a structure to exploit?
  - Scooping
  - Grids of points

Reconstruction Context

- Extension of classical Laplacian partial differential equation used to solve heat flow etc
- Now possible with new numerical computation techniques on large sparse overdetermined systems of linear equations
- Adds capabilities to the classical system
  - Local maxima inference
  - Inconsistent data conflation
ODETLAP Point Reconstruction

- Solve an overdetermined variant of a Laplacian PDE.
  - Known pts: \( z_{ij} = h_{ij} \)
  - All pts: \( 4z_{ij} = z_{i-1j} + z_{i+1j} + z_{ij-1} + z_{ij+1} \)
- Easily processes 400x400 arrays of elevation posts in Matlab (160,000 unknowns)
- Process larger arrays with Page-Saunders technique

ODETLAP on Larger Cells

- We could go to Page-Saunders if there is interest.
- My masters student John Childs did this in 2003, before Geo*.
- Goal: several-thousand-square cell.
Four Matlab Interpolation Techniques on Nested Square Contours

This difficult example was chosen to illustrate all these methods’ limitations.

ODETLAP on Nested Squares

Surface now looks much better. Can tradeoff accuracy vs smoothness.
Terrain Test Data

Extracted from level 2 DEMs

Elevation range

Hill1

RPI / Geo* / NGA & DARPA Oct 1-2 2007
Mtn3

Accuracy Metrics

- Since flatter cells are easier,
- Following slides and tables show
  - RMS error, meters, or
  - (RMS error) / (elevation range in the cell)
- Slope computed using Zevenbergen-Thorne algorithm used in NGA HAT.
- Slope error is always RMS degrees.
### TIN + Greedy ODETLAP Results

<table>
<thead>
<tr>
<th>Data</th>
<th>Size, bytes</th>
<th>Compression ratio</th>
<th>RMS Elev Error, m</th>
<th>RMS Slope Error, deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>hill1</td>
<td>1880</td>
<td>170:1</td>
<td>2.83</td>
<td>3.53</td>
</tr>
<tr>
<td>hill2</td>
<td>1962</td>
<td>163:1</td>
<td>4.06</td>
<td>8.06</td>
</tr>
<tr>
<td>hill3</td>
<td>1739</td>
<td>184:1</td>
<td>1.66</td>
<td>1.65</td>
</tr>
<tr>
<td>mtn1</td>
<td>1979</td>
<td>162:1</td>
<td>3.77</td>
<td>14.0</td>
</tr>
<tr>
<td>mtn2</td>
<td>2006</td>
<td>160:1</td>
<td>4.31</td>
<td>14.1</td>
</tr>
<tr>
<td>mtn3</td>
<td>2004</td>
<td>160:1</td>
<td>4.58</td>
<td>13.3</td>
</tr>
</tbody>
</table>

### TIN+Greedy Elevation Accuracy

The diagram shows the relationship between compressed size and RMS elevation error percentage over the range of compressed sizes. Each data set, labeled as Hill 1, Hill 2, Hill 3, Mtn 1, Mtn 2, and Mtn 3, is represented by a distinct line color and marker style, demonstrating how the compression ratio affects the error percentage at different compressed sizes.
**TIN+Greedy Slope Accuracy**

Compressed Size vs. Error

- Hill 1
- Hill 2
- Hill 3
- Mtn 1
- Mtn 2
- Mtn 3

**TIN+Greedy Elevation Comparison**

**Mtn2 Dataset**

- mtn2 Original Surface
- mtn2 Reconstructed Surface

7641 bytes => 42:1 compression ratio
Zevenbergen-Thorne Slope Comparison
Mtn2 Dataset

7641 bytes => 42:1 compression ratio

---

Even Better Slope Representation

- **Idea:** Extend the ODETLAP equations directly to incorporate the original representation's vector gradient at critical points, instead of inferring slope from adjacent elevations.

- **Status:** being designed.
Different Point Selection Strategies

- Previous slides used TIN+greedy
- That can be made to allow progressive transmission of the points, by replacing bitmap coding of the (X,Y) with a bzip2 compression.
- Following slides use regular grid point selection.
- That compresses better but doesn’t do progressive transmission.

---

Regular Grid ODETLAP Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Points</th>
<th>Compressed Size</th>
<th>Compression Ratio</th>
<th>Elev RMS (m)</th>
<th>Slope RMS (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hill1</td>
<td>529</td>
<td>306</td>
<td>1046:1</td>
<td>9.63</td>
<td>4.32</td>
</tr>
<tr>
<td>Hill2</td>
<td>1600</td>
<td>807</td>
<td>397:1</td>
<td>9.98</td>
<td>6.54</td>
</tr>
<tr>
<td>Hill3</td>
<td>225</td>
<td>172</td>
<td>1860:1</td>
<td>9.71</td>
<td>3.04</td>
</tr>
<tr>
<td>Mtn1</td>
<td>4489</td>
<td>2194</td>
<td>146:1</td>
<td>9.66</td>
<td>10.13</td>
</tr>
<tr>
<td>Mtn2</td>
<td>4489</td>
<td>2027</td>
<td>158:1</td>
<td>9.95</td>
<td>10.34</td>
</tr>
<tr>
<td>Mtn3</td>
<td>4489</td>
<td>2013</td>
<td>159:1</td>
<td>9.91</td>
<td>9.85</td>
</tr>
</tbody>
</table>

(Our second ODETLAP point insertion strategy)
### Regular Grid ODETLAP Accuracy

**Compressed Size vs. Error**

- **% RMS Elevation Error/Range**
- **Compressed Size (Bytes)**

- **Hill 1**
- **Hill 2**
- **Hill 3**
- **Mtn 1**
- **Mtn 2**
- **Mtn 3**

### Merged Point Selection Strategy

- Start with regular grid
- Greedily add more points,
- Use overlapping local grids, like with differential geometry coordinate frames.
- **Status**: being designed.
Missing Data Fillin

- ODETLAP can fill in missing circles with r<=100.
- Slopes are continuous across the boundary.
- Contours are realistic.
- Next slide compares ODETLAP to 3 Matlab methods.

Fillin Comparison

- ODETLAP
- Laplacian
- Thin plate
- Matlab nearest
- Matlab linear
- Matlab cubic
Packaging ODETLAP

- Current process is a group of programs combining Matlab, bzip2, C++.
- Being packaged into a unified system for distribution to NGA.
- We can complete this when needed.

More Terrain Representations

- Scooping still has the greatest longterm potential.
- Could use RPI’s BlueGene/L, the 2nd fastest computer in a university setting in the world. 
  (source: http://top500.org/lists/2007/06)
- Problem: It doesn’t run Matlab.
Goal 2: Smugglers and Border Guards (aka Siting & Path Planning)

- **Terrain**
- **Parameters:**
  - Observer height
  - Target height
  - Radius of interest
  - Intervisibility?

**Siting program**

- **Observer positions**
- **Joint viewshed**

---

**Multiobserver siting steps**

1. Compute approximate visibility index of every possible observer.
2. Compute exact viewsheds of the best.
3. Greedily insert potential observers into the final set of observers, maintaining a bitmap of the cumulative viewshed.
4. Intervisibility => insert only visible observers.

**Key:** fast bitmap operations allow hundreds of observers to be sited with hi-res viewsheds.
Sample Viewsheds

Note the level of detail

Viewshed uncertainty

Hue indicates elevation

Visible
Possibly hidden
Hidden
Probably hidden
Observer
**With or w/o intervisibility**

- **Intervisibility enforced**
- **No intervisibility required**

Color -> elevation; Black -> hidden.

---

**Siting Toolkit by ESRI**

- **ArcGIS DLL** toolkit: an operational class configurable to perform siting simulations on any platform with a C++ compiler.
- Includes an ArcMap command application on Windows, to demonstrate its capabilities.
- Both are functional and scalable.
- Marquee Tool:

```
Spatial Analyst Layer: dataset_views.tif
```

---
ArcGIS DLL Dialog Box

Path Planning (Smugglers)

- Find cheapest path between source and goal.
- Cost metric is not simply path length:

\[
c = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad \text{(Distance)}
\]

\[
\cdot \left(1 + \max \left(0, \frac{\Delta z}{\sqrt{\Delta x^2 + \Delta y^2}}\right)\right) \quad \text{(Climbing costs)}
\]

\[
\cdot (1 + 100\nu) \quad \text{(BIG penalty for being seen)}
\]
Path planning algorithm

- Designed for hi-res, say 1000x1000, maps.
- Impossible to form the $10^6 \times 10^6$ cost matrix.
- Use A* to search for initial feasible, good, path.
- Iterate to optimize it.
- Doesn’t hang up on local optima.
- Compute many paths to evaluate compression throughout the terrain.
- Note how complex our paths are.
- Video: multipath.wmv

Many Paths on Each Dataset

- hill1
- hill2
- hill3
- mtn1
- mtn2
- mtn3
Many Paths on Hill1

Computed between 50 pairs of random start/end points

Many Paths on Mtn1
Many Paths on Mtn3

Path traversal video

- RPI-path-planning.wmv
**Alternate Terrain Representation Evaluation Using Path Planning**

- **Q:** Our alternate compressed representation has a good RMS elevation error. However, is it good for more sophisticated operations, like path planning?
- If we compute a smugglers path on terrain stored in our alternate rep, how good is it really?
- It’s not important if a computed path is very different from the optimal path, *provided* that its true cost is not much more expensive than the optimal path.

**Smugglers Path Evaluation of ODETLAP**

- **Size:** size of compressed dataset in bytes. Original binary size=320KB.
- **Incr. cost:** extra cost of optimal path computed on compressed dataset and evaluated on original dataset compared to optimal path computed on original dataset.

<table>
<thead>
<tr>
<th></th>
<th>Data</th>
<th>Size</th>
<th>Compr. Ratio</th>
<th>Incr. Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>hill1</td>
<td>1763</td>
<td>182:1</td>
<td>5.5%</td>
<td></td>
</tr>
<tr>
<td>hill2</td>
<td>1819</td>
<td>176:1</td>
<td>6.1%</td>
<td></td>
</tr>
<tr>
<td>hill3</td>
<td>1607</td>
<td>199:1</td>
<td>4.4%</td>
<td></td>
</tr>
<tr>
<td>mtn1</td>
<td>1925</td>
<td>166:1</td>
<td>19.2%</td>
<td></td>
</tr>
<tr>
<td>mtn2</td>
<td>1884</td>
<td>170:1</td>
<td>18.2%</td>
<td></td>
</tr>
<tr>
<td>mtn3</td>
<td>1946</td>
<td>164:1</td>
<td>17.0%</td>
<td></td>
</tr>
</tbody>
</table>
Path Planning for Road Construction

- **Goal:** Construct an optimal road connecting two points.
- **Allowed:** Material removal and deposition.
- **Constraint:** Max allowable slope is bounded.
- **Objective function:** Amount of material moved.
- **Method:** A*
**Key Differentiating Factors – Smugglers and Border Guards**

- Can site hundreds of observers on the terrain.
- Observers' radius of interest may be several hundred pixels.
- Visibility and navigation tool
  - optimizes a sophisticated objective function consisting of the path's total distance, the amount of uphill travel, and the distance spent in sight of any observer.
- Paths may be thousands of pixels long.
- Validates ODETLAP:
  - Good RMS elevation error *but wait, there's more!*
  - Can be used accurately to site observers and plan paths.

---

**Hydrology Problem**

Just starting this Phase II task in the proposal.

- Compute streams from terrain, ...
- Assuming water flows from each cell to lower neighbors.
- **Problem:** many cells are local minima trapping flow
- **Why?** Data errors; insufficient sampling
- **Effect:** No long streams.
Common Solution

- Simulate gradually filling in the local minima until the water flows over the edge.
- *But:* This is slow

  - *Real-world implementation of this technique with the Taum Sauk reservoir; before and after.*

Our Techniques

- Identify sinks & watershed boundaries with our very fast connected components program.
- Solve for water flow as a sparse system of linear equations.
- Invert elevations and solve for *ridge rivers*.
- Merge watersheds not blocked by significant ridges.
- Input ridge and stream points to ODETLAP.
Fast Connected Components

- **Original application:** Small block of concrete is stressed to failure while being CAT-scanned in Brookhaven synchrotron. First breaks into a few large blocks, ...
- Need the 3D structure to understand failure.
- Compute the connected components of the thresholded 1000x1000x1000 CAT scan.
- **Next slide:** slices in 3 different directions look quite different.

Slices in 3 Directions of Cracking Concrete Block
Method

- Good implementation of union-find.
- Fundamental data structure is a 1-D run of solid voxels. (assumes data coherence)
- Carefully designed, small and fast.
- Form connected components with a few passes through the data.

Largest Test Run

- **Input**: 1024x1088x1088 = 1,212,153,856 voxels, 50% empty.
- **Output**: 4,539,562 components, averaging 4.5 runs.
- **Largest component**: 2993 runs, volume 23675.
- **Many components**: only one run and two voxels.
- **Implementation**: 2GHz IBM t43p laptop, linux, gcc.
- **Virtual memory used**: only 340MB
- **Elapsed time**: 51 CPU seconds.
- **Times scale down**: ≈ 0.1 secs for 100x100x100.
- **Largest 2D test**: 19000x19000
Merging Watersheds not Blocked by Significant Ridges

Before merging

After merging

=> Larger more realistic watersheds and drainage networks
Summary

- Represent terrain in 1% of original binary space with compression ratios of 80:1 to 500:1 with 10m elevation and 5-10 degree slope error.
- Site multiple observers ("border guards") and then plot smugglers paths to avoid them.
- Compute on the compressed terrain.
- Modify terrain to improve hydrology.