Displacement Mapping
A New Method to Achieve Realistic Geo-Specific Feature Densities

Brett Chladny

RENAISSANCE SCIENCES CORPORATION

John Femiani

ASU

The Problem/Need

- 3D features provide visual cues required for low level flight training
  - Assessing speed and height above terrain
  - Motion parallax created by vertical edge
  - Improved realism

- Limited numbers of features exist in modern day visual simulation
  - Creating feature rich databases is expensive
  - Sending every feature down the graphics pipe can be inefficient

Goal

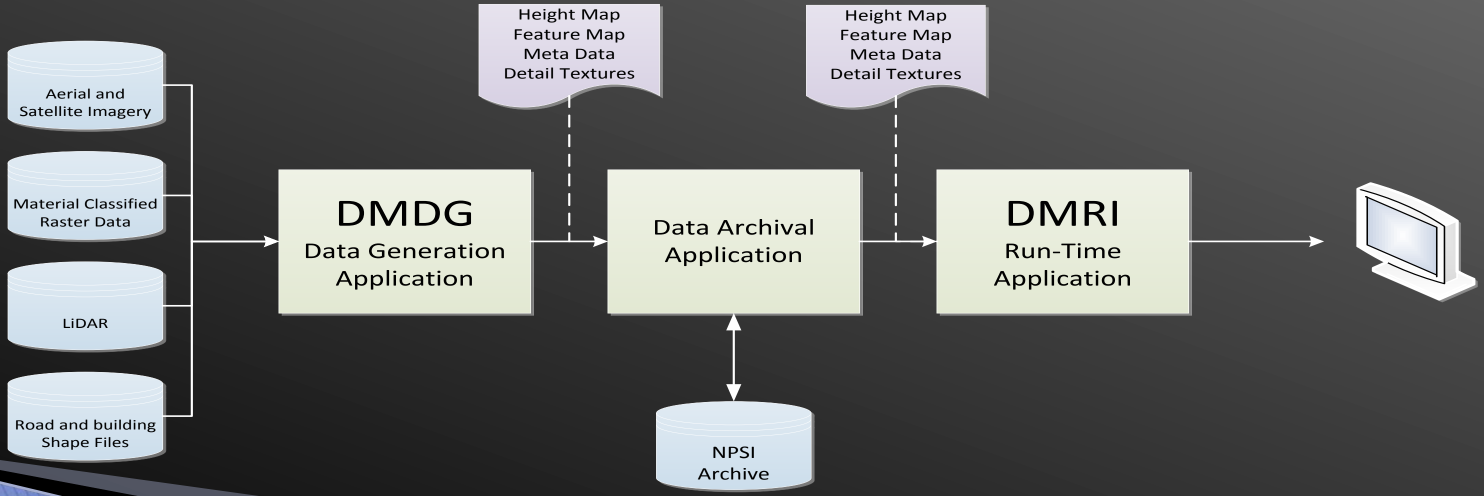
- To develop a method for adding unlimited numbers of feature to visual simulations
  - Automate feature extraction from commonly available data sources
  - Deform the terrain's surface to give the ***appearance*** of unlimited numbers of features
  - Integrate technique into modern day image generators with minimal impact to the IG performance

Solution

- Develop automated way to extract features from commonly available data sources
  - Feature extraction from imagery alone is possible, but difficult
  - Optain different types of information from different types data
  - Produce **plausible** results
- Develop method for visualizing large numbers of features
  - Displacement Mapping
    - Re-tessellate the terrain surface on the GPU
    - Deform the terrain surface to provide the **appearance** of identified features
  - Polygonal approach
    - Very efficient !!!

Approach

- Pipeline solution
  - Generate displacement and shape file data
  - Archive data as refined source data
  - Develop runtime solution that makes use of data

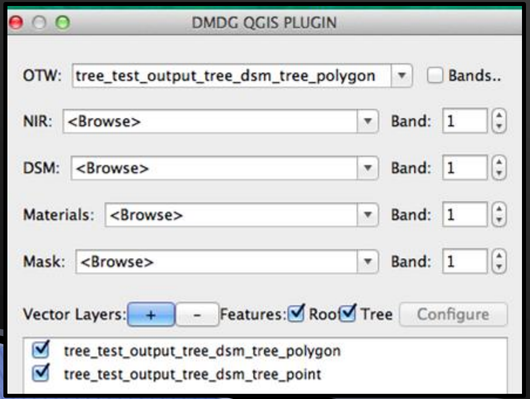


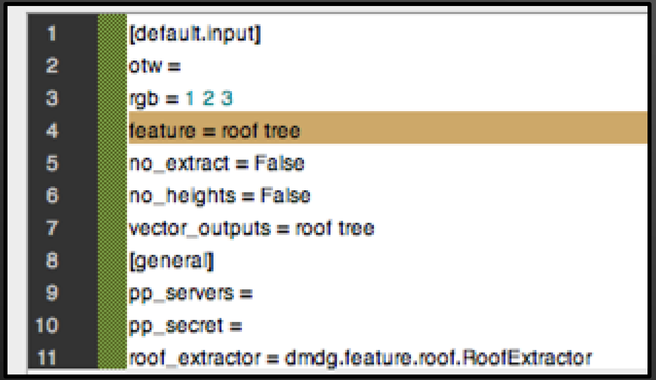
DMDG

Displacement Map Data Generator

What is DMDG?

- Python (mixed with C / cython) package
- Command line tool
- Graphical plug-in to QGIS
- Can be run locally or remotely on a Linux cluster



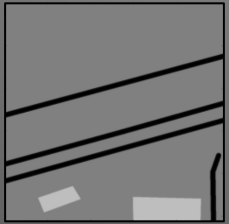


DMDG input

- Data fusion from multiple sources
  - Photopic imagery (out-the-window RGB)
  - NearIR imagery
  - Vector (street maps, building foot prints, etc.)
  - DSMs (from LiDAR, stereo pair imagery, etc.)

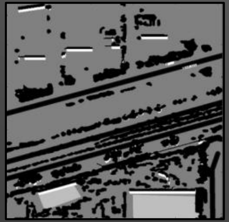
Data Fusion

- Each type of input provides different information:
  - Constraints on where features can be placed
  - Cues to guide model fitting and feature detection
- Leverage all of the data available
  - Data layers are used to modify a one-per-feature-type “guidance” image.



Vector data adds constraints based on roads and hints from building footprints.

Start with an OTW image



Shadow & vegetation analysis gives additional constraints

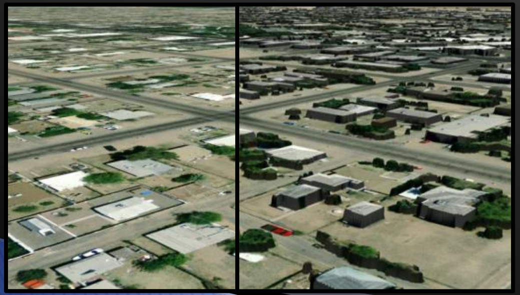
DMDG aligns and completes guidance image to match OTW pixels based on RGB data

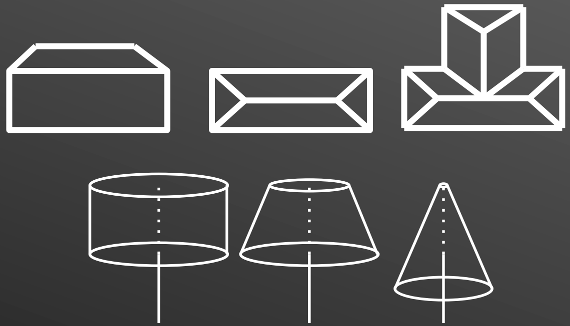
No Feature | Improbable Feature | No Indicator | Probable Feature | Definite Feature

DMDG output

- Max height of features at each pixel

- Point features





Processing Time

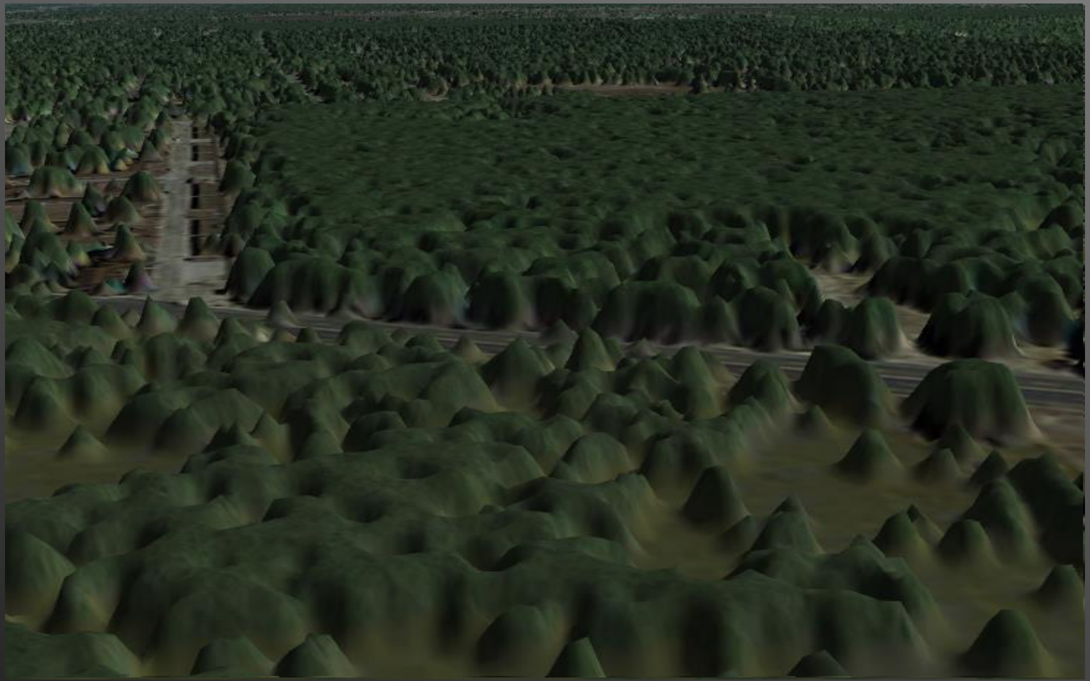
- DMDG is usually run on a cluster of computers
- Approximately 30s per km^2 at 1mpp
- Time varies by image contents & resolution
- Less then 24 hours for 300 square miles of 25cm imagery
- Looking into GPU acceleration

FIL

Feature Injection Library

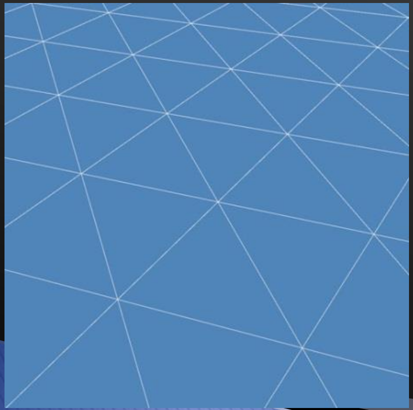
History of Displacement Mapping

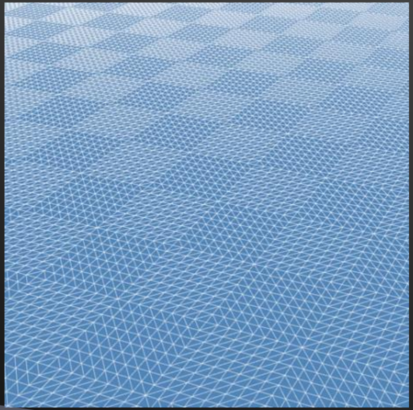
- Polygonal lighting
- Bump Mapping
- Parallax Mapping
- Displacement Mapping

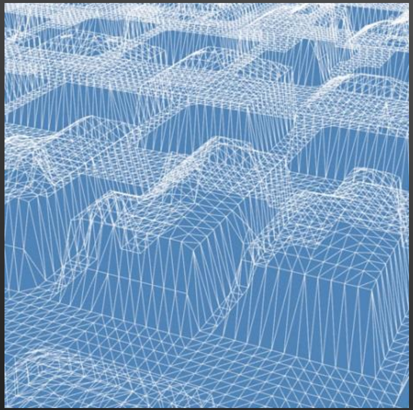


Displacement Mapping Technique

- Start with normally tessellated terrain skin
- Re-tessellate terrain skin on GPU
- Move resulting vertices based on displacement map
- Render new terrain skin to reveal both the terrain surface and the ***appearance*** of features







Comparison with Traditional Methods

- Traditional rendering approach
  - Each feature is represented by its own set of geometry
  - More features = longer rendering time

- Displacement mapping
  - Consistent rendering performance
  - ***Appearance*** of features created by deforming terrain's surface
  - Terrain tessellation based on distance not feature density
  - More features = same rendering time


1000 Feet


500 Feet


100 Feet

Polygonal Rendering Technique

- Scene graph independent self contained rendering solution
- Render features relative to the observer (supports flat and round-earth)
- Load and render features efficiently!

Polygonal Model Generation

- Geo-specific footprints – from DMDG
- Geo-specific elevation – from DTED used to build database
- Geo-specific roof colors – from OTW imagery
- Group into tiles that are 1/10,000 of a geo-cells
- Store data in files on disk

Polygonal Model Rendering

- Supports both round and flat earth
- Can be integrated into any OpenGL based IG
- Reads data files disk and constructs features in parallel thread
- Constructs OpenGL primitives in VBOs
- Uses less then 1 ms per frame to upload data to the GPU
- Sets up pipeline for rendering each frame
- Directly renders features into active draw context
- ***Plausible*** visual result with impressive performance


Frame Rate:59.97
DMRI
Reload the shaders
Toggle Features
Toggle Color Bands
Toggle Morphing
Latitude 32.87596
Longitude -111.75697
Altitude 1494
Time of Day 12.3
Displacement 0.00
Scene LOD 1.00
Blend Range 15000
Field of View 60.0
Near Clip 10
Far Clip 100000
Velocity in Mph: 0
Altitude in feet: 1494
Tiles Rendered: 241
Tiles Culled: 774
Num Triangles: 11208042
Lat: 33.2162
Lon -117.2652

Hybrid Solutions

- Near-field: Polygon Features
- Far-field: Displacement mapping
- Transition zone
  - Polygonal features fade out using alpha
  - Displacement amount starts from zero
- Displacement mapping only, polygonal features only, and hybrid solution all possible

Use Cases

- Fixed wing Nap-of-the-Earth flight training
- Rotor wing flight training
- UAV and camera operator training
- Search and rescue training
- Targeting pod training
- Mission rehearsal

On to the Demo…

Acknowledgements

- Dr. John Femiani – Arizona State University
- NAVAIR SBIR program for providing the funding for this research*
- SBIR Topic N102-116: “Geospecific Displacement Maps for Real Time, Stereoscopic Training Simulation”

*Disclaimer: the views that have been presented are those of the authors and do not necessarily represent the views of DoD or its Components.

Brett Chladny
bchladny@rscusa.com
Phone: 480-374-5068

Please complete the Presenter Evaluation sent to you by email or through the GTC Mobile App. Your feedback is important!