

TerraVision II: *Using VRML to Browse the World*

Yvan G. Leclerc <leclerc@ai.sri.com>

Martin Reddy <reddy@ai.sri.com>

Lee Iverson <leei@ai.sri.com>

SRI International



TerraVision I

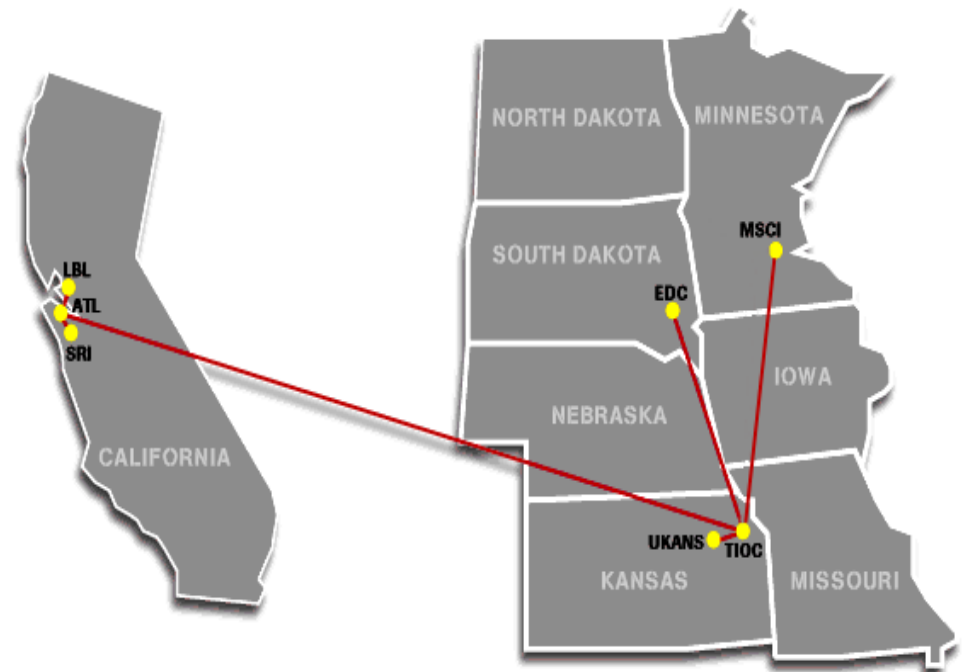
- Overview of TerraVision I
- Features of TerraVision I
- Optimisation Issues
- Video Footage

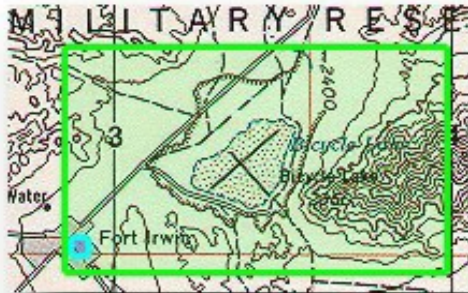
TerraVision II

- Overview of TerraVision II
- Potential Applications
- Why and How to Use VRML
- Problems Encountered with VRML
- Conclusions



- Interactive, distributed terrain visualization system
- The primary application of the MAGIC project
- Terrain data reside on Distributed–Parallel Storage Systems (DPSS) on the wide–area MAGIC network
- Goal = to be able to visualize tens to hundreds of gigabytes of data interactively and from arbitrary vantage points

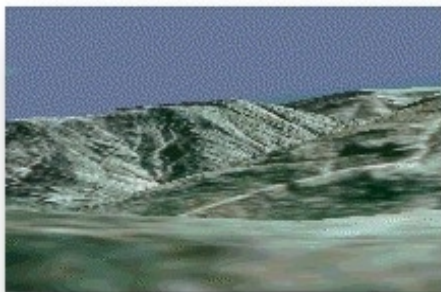




MAP VIEW : *shows the extent of the current dataset and the view frustum overlaid on a map.*



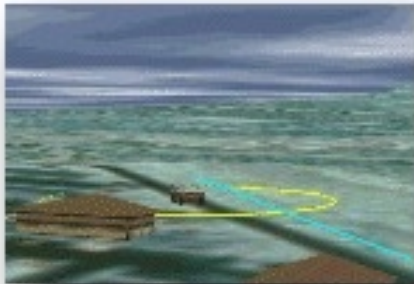
2-D Pan & Zoom : *overhead view of imagery, registered with the map view. Users can pan and zoom over imagery.*



3-D Fly Throughs : *view terrain in 3-D at interactive rates (10–30 Hz). User's viewpoint is projected onto the registered map.*



Atmospheric Effects : *user can modify effects such as time-of-day, fog, lighting, texture-mapping, etc.*



Buildings & Live Vehicles : *building and vehicle models can be superimposed on the terrain. Vehicles can be registered to a GPS signal.*

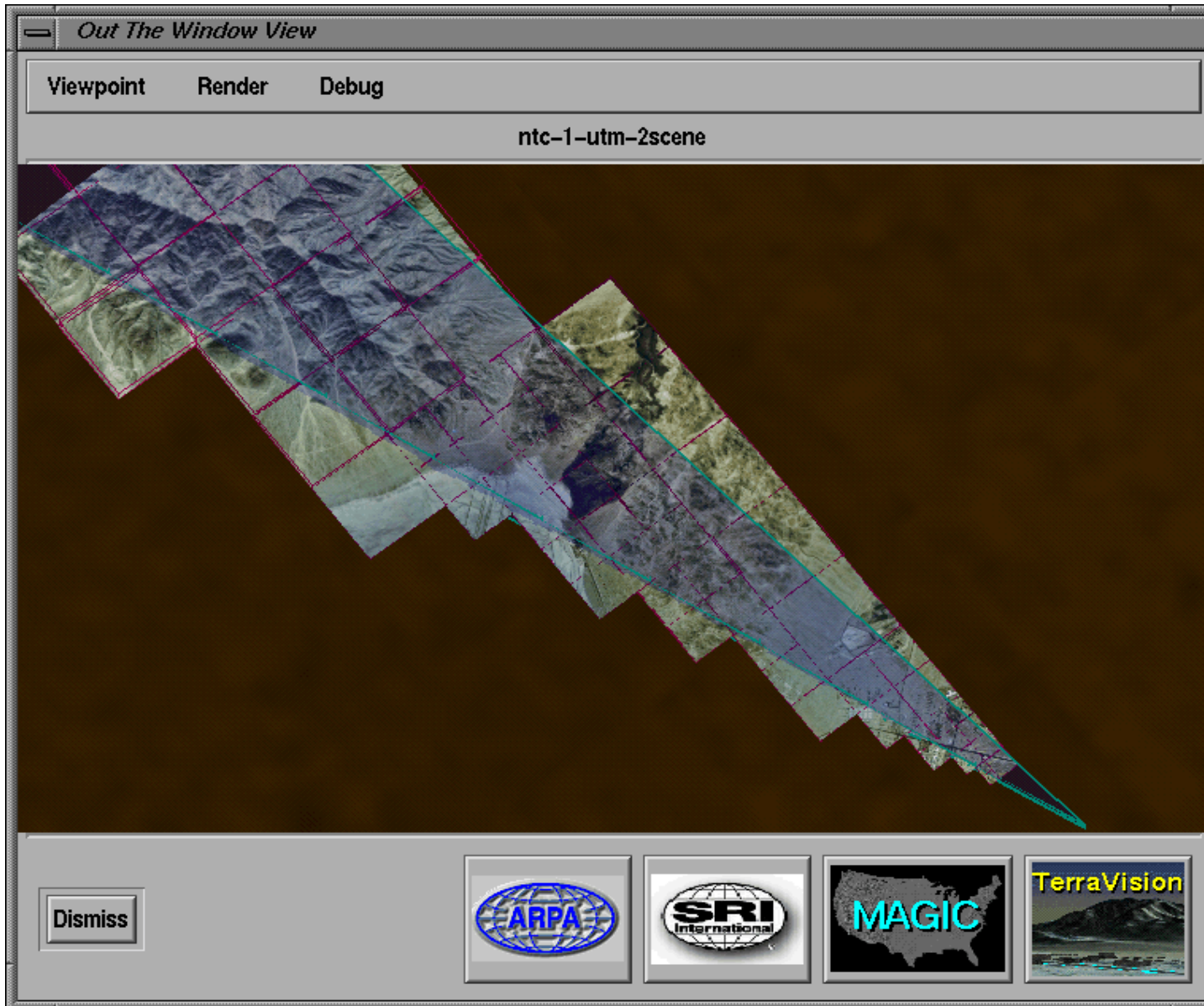


Input Devices : *available user interfaces include: mouse, SpaceBall, CAVE, Head-Mounted Display, and AVS module.*



TerraVision can visualize massive databases (many GBs), so optimisation techniques are critical to maximise network bandwidth, processing power, and local disk space, e.g.

- **Level of detail (LOD)** : *a multi-resolution representation of the terrain data is generated by segmenting into a number of rectangular tiles at various resolutions. TerraVision then uses the lowest resolution possible for good image quality.*
- **Culling of Non-Visible Tiles** : *terrain tiles which are not visible are never transmitted over the network, or rendered to the screen.*
- **Prediction and Pre-Fetching** : *TerraVision predicts where the user will be in the near future and prefetches the terrain data for that region.*
- **Efficient Caching** : *terrain tiles are cached locally (LRU) so that they need not be retransmitted on each frame.*



TerraVision screenshot showing the effect of the view-dependent level of detail algorithm and the culling of non-visible tiles.



TerraVision screenshot showing the same scene as the previous slide, but from the user's viewpoint.

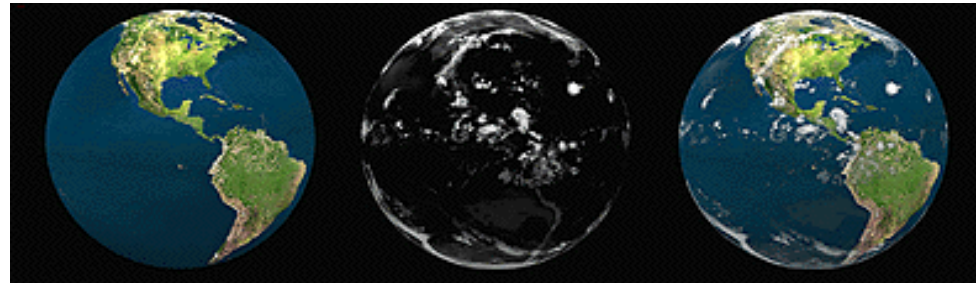


Play Video Sequence...



TerraVision II under development and will differ from TerraVision I in the following major respects:

- VRML 2.0 used to store all terrain data**
(data should be viewable by TerraVision and a standard VRML browser)
- Multiple images can be viewed and blended together**



- Visualize the entire Earth: not just small, flat regions**
- Cross-platform portability (OpenGL / C, pthreads)**



Example One: BROWSING THE WORLD

- User sees 3–D view of the earth and zooms into Menlo Park, CA.
- As she approaches, TV–II informs her that several new data are available, including maps, infrared imagery, vector road data, etc.
- She chooses which of these data to view (blended or overlaid)

- Certain data are not available so she uses an on–line HTML form to let her create the data and place it on the nearest DPSS
- She can fly back when then data is available or wait for it to appear progressively

- As she approaches the ground, buildings and vehicles appear
- Some of the buildings are 3–D anchors, pointing to HTML pages

BROWSE

CREATE

LINK



Example Two: FIGHTING A FOREST FIRE

- User wants to visualize a forest fire happening at that time in order to plan evacuations and fire fighting strategies
- He needs the terrain dataset urgently so requests the use of several dozen high-speed workstations to parallelize the creation of his data
- The terrain data has pointers to dynamic imagery which are created in real-time as an airplane flies over the site
- TerraVision can update the view of the scene as each new dataset becomes available
- The user can request that all images created to date be animated back to see the progression of the fire from various

PARALLEL

UPDATE

ANIMATE



Why use VRML to store TerraVision's terrain data?

- data viewable by any standard VRML browser**
- pervasive (and growing) support exists for VRML**
- VRML provides level of detail support**
- ability to create hyperlinks within the world**
- platform independent**
- VRML 2.0 is on the path to becoming an ISO standard**

How to represent a view-dependent multi-resolution terrain hierarchy in VRML?

- each terrain tile is represented as a separate VRML file
- other VRML files can be used to "glue" all of the tiles into a level of detail hierarchy
- e.g. use a quad-tree like structure : a single tile has an LOD node which replaces itself with four (or more) higher resolution tiles when the user gets close enough to it.





Level of detail (LOD) restrictions in standard VRML viewers:

- Tearing can occur between tiles of different LOD**

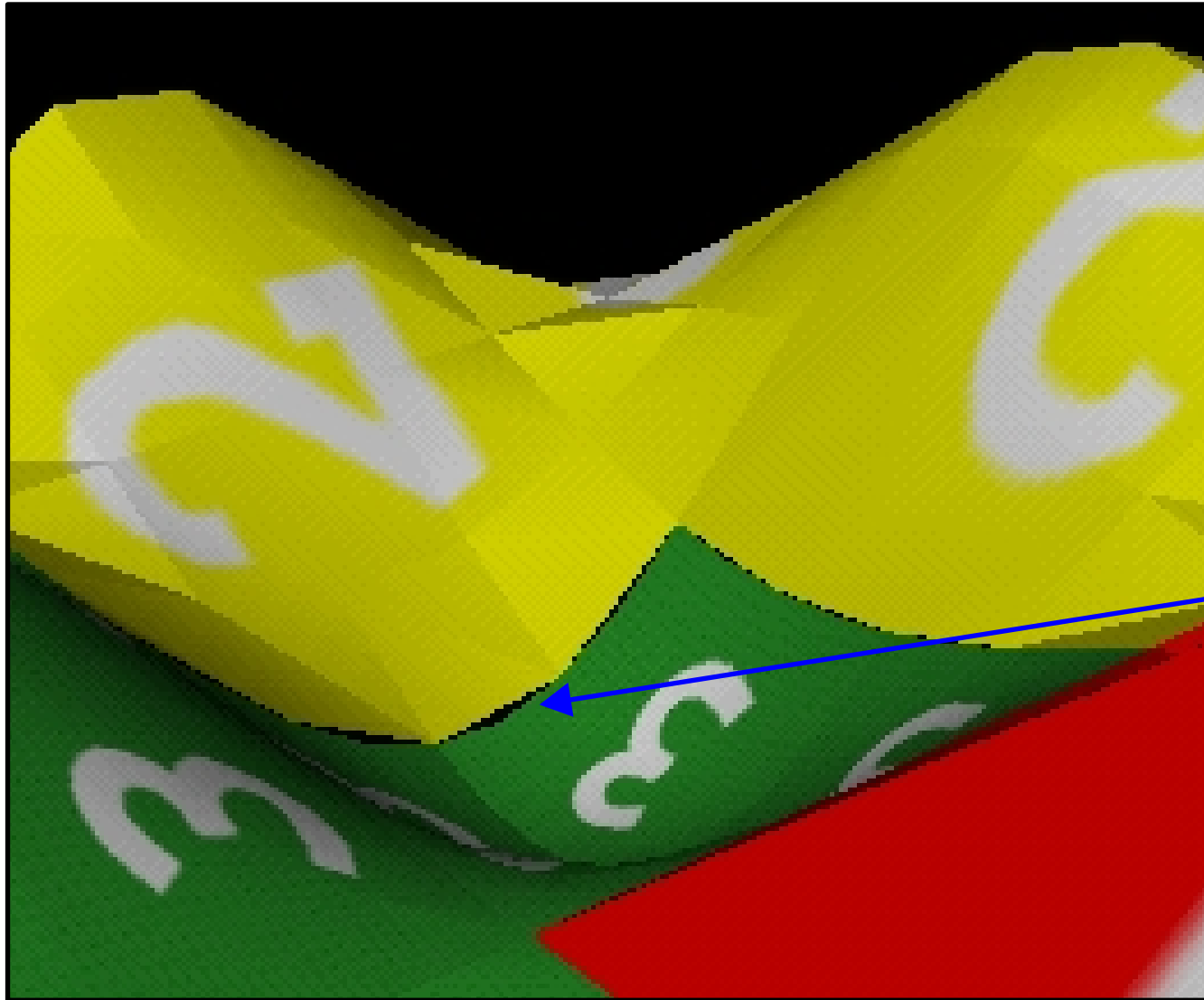
Currently no mechanism to smooth LOD between adjacent tiles in VRML

- VRML LOD model is generic and proves unmanageable for terrain hierarchies with more than about 3 LODs**

Circumvented by segmenting the hierarchy into a chain of scenes in which each scene has only 3 LODs, with the last detail level offering a hyperlink to open a new scene with the next 3 LODs.

- LOD switching distance is arbitrary**

The optimal distance varies as a function of screen size, terrain complexity, viewpoint, etc.



Tearing can occur between adjacent tiles at different levels of detail in a VRML browser



Imagery support restrictions:

- Cannot easily switch between different image datasets**

Each tile is an independent file and events cannot be passed between these, so there is no way to globally change the imagery for all tiles.

Solution = don't try! Provide the information for TerraVision, but VRML will ignore this.

- No support for dynamic imagery – reloading new data**

VRML works by loading a world and then running it. There is no concept of updating data if it changes on the server. We therefore don't provide support for dynamic imagery in the VRML browser



- VRML uses single precision floating point numbers**

We need double precision to model the Earth to an accuracy in the order of centimeters or less. Overcome this by implementing a hierarchy of Local Coordinate Systems which are transformed into the Global Coordinate System.

- Where is up?**

Current VRML navigation metaphors assume that the Y-axis is up in the world coordinate system. When navigating a globe, "up" will continually change. We need new navigation techniques for globes.

- Altitude-dependent velocity**

Travelling at 200 mph on land is relatively fast, but in space it's a crawl! We need navigation techniques specific for large terrain visualizations.



- TerraVision II to use VRML as the file format for all terrain data
- Goal is for standard VRML browser to be able to browse TerraVision's datasets as much as possible
- Good: VRML widely used and offers a format with pervasive tools and browsers across multiple platforms
- Not So Good: more direct support is required in VRML for representing geographic databases

URL for TerraVision = <http://www.ai.sri.com/~magic>