**Creating Geospecific 3-Dimensional Databases for Real-time Visual Simulation**

**Part I: Introduction**

Logically, the first question one asks is, "What are you talking about?" In essence what we are referring to is a 3-dimensional virtual world, only there are a few other key words; a) Geospecific: correlated to the real world both in size and placement, and b) Real-time: updated at 60 frames per second (or thereabout), good enough so a pilot or any other human would not be able to detect any apparent "jitter" in the visuals.

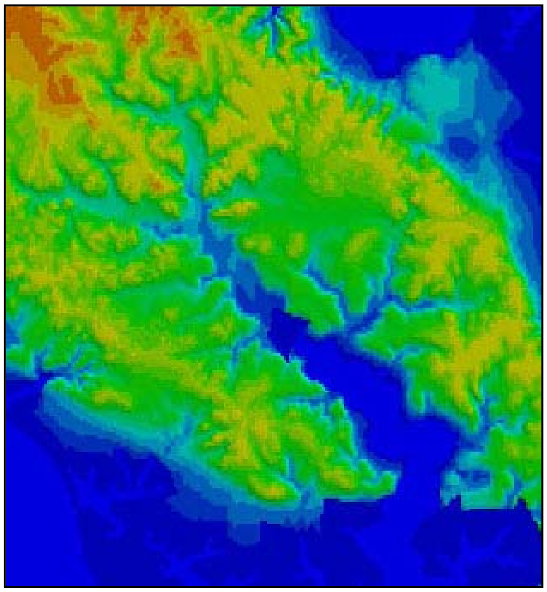
The idea for this St. Mary's Project stems from the development of flight simulators for the Navy at Naval Air Systems Command, in Patuxent River, Maryland. Our database will be different from those used in simulators in that it has a much more local focus. Our database will not cover such a large area, but this will allow us to densely populate the database with a number of cultural features. In other words, more buildings, trees, light posts, trashcans, etc.

**Part II: Making a Visual Database**

So how do we make it? The database consists of a number of different datasets that are manipulated and merged together: 1) raster imagery, satellite and aerial photographs of the terrain, 2) elevation data, data used to create the terrain skin, the land the imagery will rest upon, 3) 3-dimensional models, similar to CAD drawings they are representations of buildings, light posts, and anything else 3-dimensional that will exist in the database, 3) vector data, data comprised of point and line features which define locations for everything from the placement of a headstone to roads and water features, and 4) digital photographs of all of the 3-dimensional cultural features, applied to the 3-dimensional models as textures. Once we've prepared all of the data, we simply import it into a database tool and with a bit of massaging, we build the complete database.

**Raster Imagery**

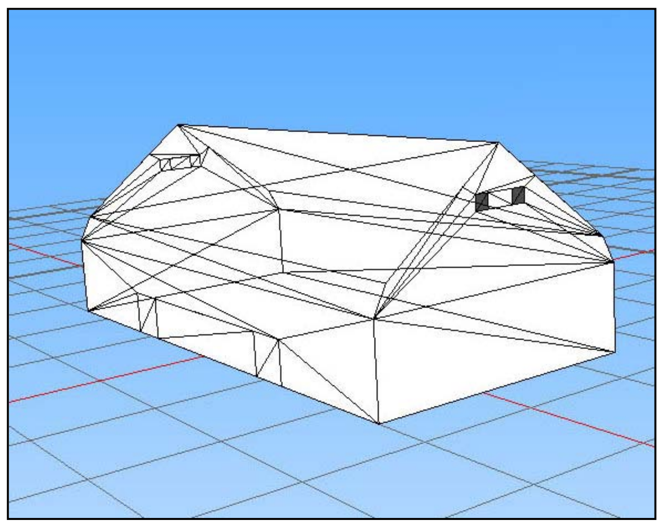
So let's start with raster imagery and how it is processed. As of now the only imagery in use is Digital Orthographic Quarter Quadrant data acquired from the USGS, United States Geological Survey. This dataset is multispectral (color) imagery at 4ft resolution – each pixel represents a 4ft x 4ft square area on the ground. The data was derived from a blue, green, infrared image, initially used to identify vegetation. As this was not our focus the data had to be run through a process which converts red, green, infrared to RGB (red, green, blue), the visual spectrum. The data was also georeferenced to the UTM WGS84 coordinate system, whereby each pixel in the image was assigned a real-world location. The picture above is a screenshot of the database with nothing but imagery applied to the terrain skin. All imagery manipulation of this nature is performed in Erdas Imagine, a GIS software tool.



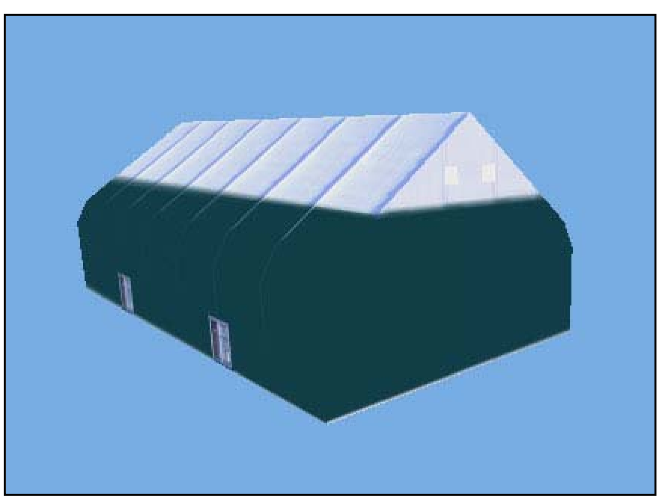
**Elevation Data**

The elevation data we use is called DTED, Digital Terrain Elevation Data. It is acquired very much the same way sonar is used to detect objects in water. A satellite passes overhead and takes a "grid posting" every 3 arc seconds or approximately 100m. Because of the large distance between measurements abrupt changes in the shape of terrain are not detected, this gives databases an unrealistically flat appearance. We are currently exploring other datasets which will more accurately depict the behavior of the St. Mary's Campus terrain. The picture to the left depicts a "rasterized" version of the DTED used. Each point is considered a pixel value and it is rendered as a picture strictly for visualization purposes.

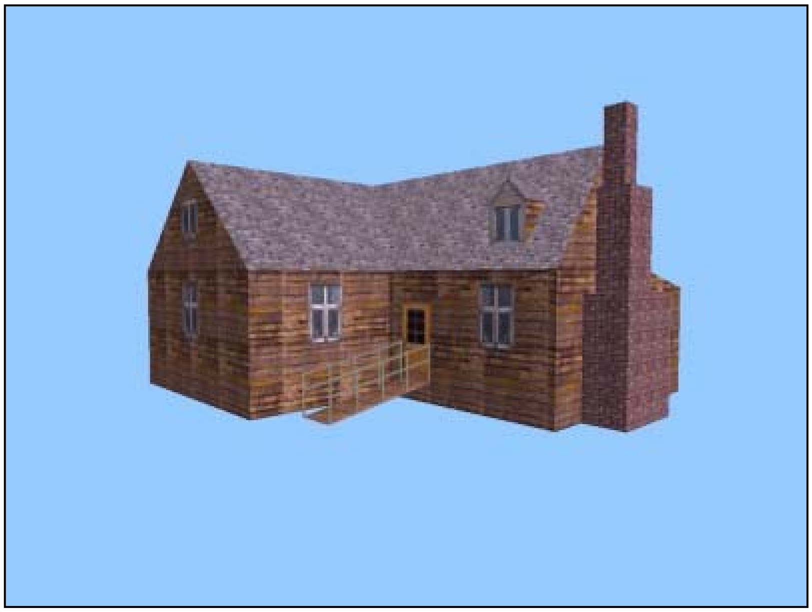
**3-Dimensional Models**



In the end 3-D models will make or break your database. They are anything that rises from the terrain, be it a tree or a building or a water tower anything which will exist in 3 dimensions in the database will be a model. If you can't make a familiar object look realistic then chances are your database will look rather cartoonish and unrealistic. However, if you can make your models look fairly close to the real thing, you can create a rather surreal experience for your audience. Basically a 3-D model is a CAD drawing with some pictures applied to it. You create a shape that resembles the object you are modeling, and then you take pictures of the objects and apply them to the faces of the object. The picture to the right is a model of the infamous BAT, now the athletic field house. The picture to the left is the same model with pictures, also called textures, applied to it.



The next page contains more examples of 3-dimensional models.



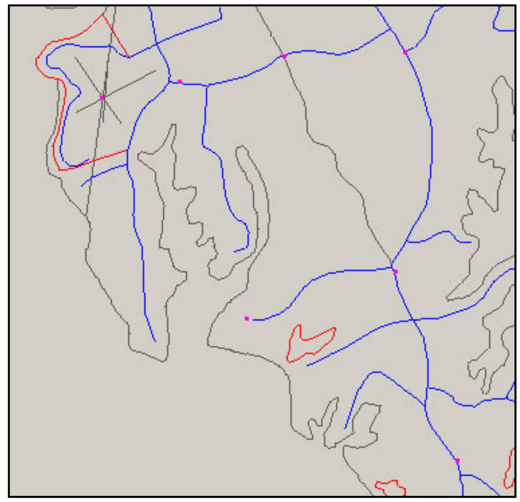
*Farthing's Ordinary*

*St. Mary's Lamppost*

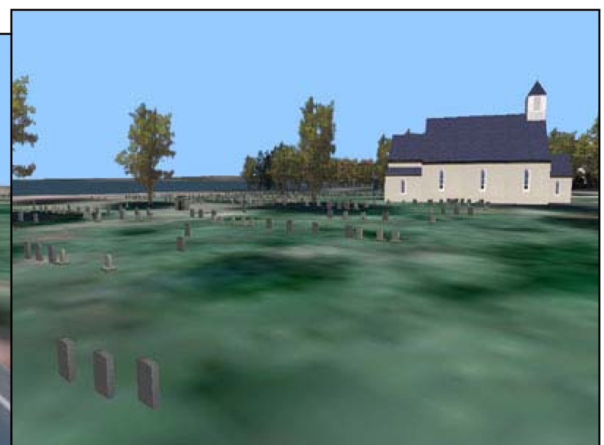
*Ivan Sterling*

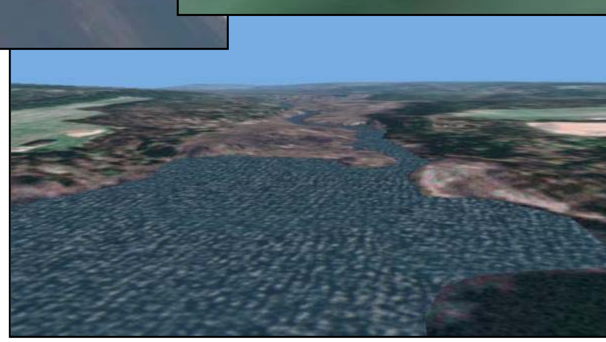
*State House at Historic St. Mary's City*

**Vector Data**



Vector data, as seen to the right, is where most of the database building process time is spent. Once we've imported our various datasets, we have to tell the tool where to place all of the objects. Generally, we bring up the raster imagery and use the building footprints to set point features for our models. However, there are a few other types of vector data including linears and areals. Linears are used for things like roads, as seen below, and powerlines. Areals are used for things like bodies of water and clusters of trees, also shown below. Sometimes you get lucky as was the case with the point features for the headstones. Another student, Dionisios Kavadias, was creating a survey of the church graveyard as part of his St. Mary's Project. Part of that survey included reporting the precise geographic location of each headstone and taking a picture to correlate with that point. The data he created was in such a fromat it could be directly imported into the database tool. As such there is a model at every point he has surveyed. As of now there are only a handfull of specific models, but that number will grow overtime. The other headstones use a generic model with no markings.

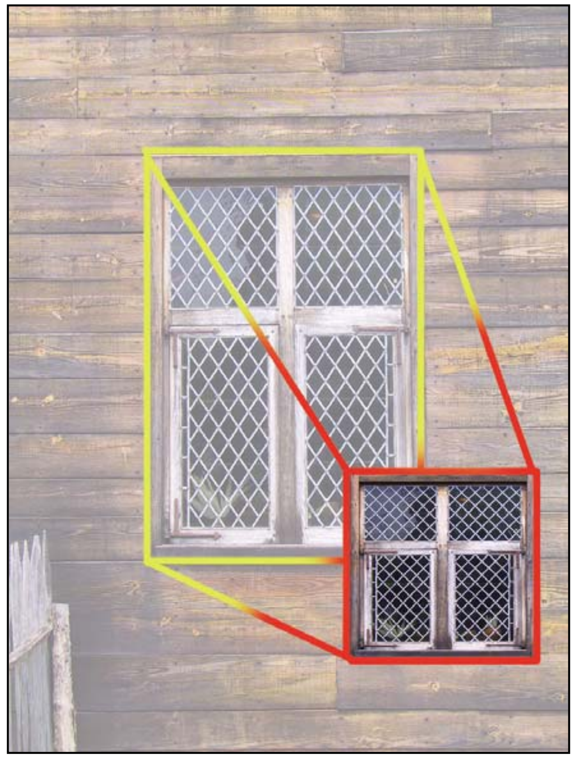




**Digital Photography**

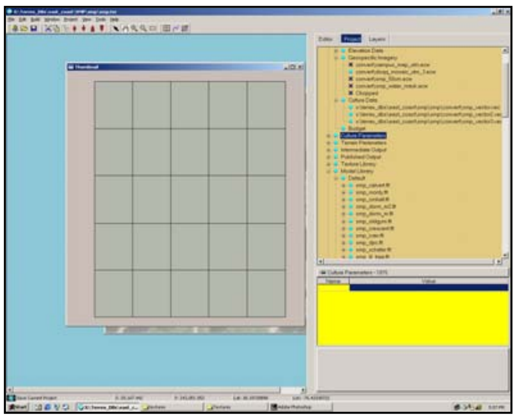
Aside from modelling, taking pictures for the models is the next most laborious task. Taking pictures in the right light, finding angles to shoot objects, and then once you have all the pictures, having to transfer them to a computer and modify each one so it may be used as a texture, it's an extensive process. The pictures in their "raw" format would be entirely too much data to process. Given say 600 KB per JPEG image, for at least 4 sides to a building, your looking at 2.4 MB of just texture. This wouldn't be all to bad except that we are dealing with hunderds if not thousands of models with the final version of the database. The processor would be able to read in the data, but at 60 frames per second the processor would be sending gigabytes of texture across an graphics pipe only designed to handle a few hundred megabytes per second. Needless to say that would be entirely to much texture. So you have to crop, make adjustments for perspective and color, and resample every image. In some cases we might generalize the materials on a building. In other words instead of using 5 or 6 large pictures of the sides of a buliding and the roof, we would make smaller textures of all the materials that make up the building. So you would have a brick texture and a wood texture and a door texture and a window texture, but they would be much smaller than the large composites. Below is the set of textures used on the Farthing's Ordinary model above.

To the right is an example of how one of the pictures above was cropped from a much larger image:

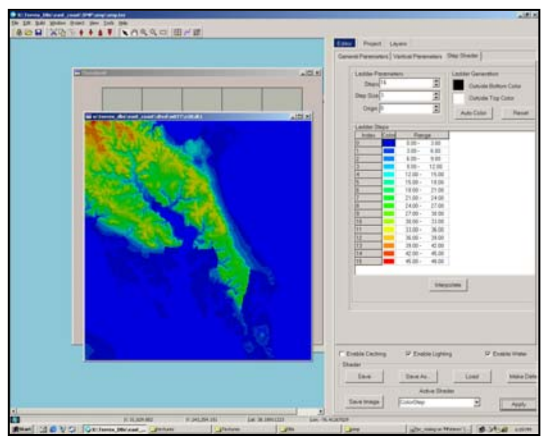


**The Database Build**

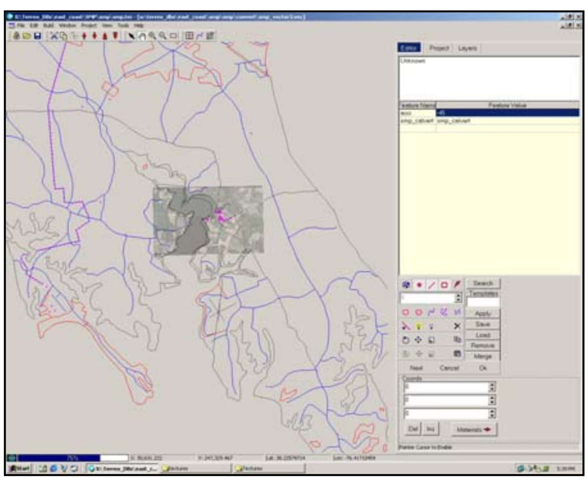
This is the final step when we bring everything together and hope it all meshes as we had hoped. There are a few screenshots below depicting several steps of the database building process.



*The Database Tool Layout*



*DTED Manipulation*



*Vector Editing*

The database build does a number of things. First it generates a terrain skin from the elevation data, in other words it creates a triangle mesh that will accurately represent the terrain and “cuts in” any vector data. Second it processes any imagery that will cover the terrain and applies it to the polygons that make up the terrain skin. Finally the build inserts all of the 3-Dimensional models we have imported and referenced in the database. Below you will find screenshots of the most recent version of the database build.