GeoMatrix™ Toolkit Technology Description

The GeoMatrix™ Toolkit, by GeoFusion Inc. is a C++ Software Development Kit that provides an object-oriented Application Programmer's Interface enabling software developers to build high-performance, interactive, full-globe digital Earth and planetary-based applications. The toolkit is a full development environment that includes modular libraries, example applications, and documentation.

GeoMatrix technology provides an increased sense of realism with world-wide datasets through features such as continuous level-of-detail while zooming, terrain morphing between resolution levels, and a multiple globes capability that allows for clouds layers, overlays, multiple planets, etc.

The GeoMatrix™ Toolkit provides increased value to existing geospatial-based visualization applications through its:

- global reference system
- space-to-street-level superior rendering quality
- high performance on low cost hardware platforms
- global terrain rendering viewable from any 3D perspective
- fully interactive control of rendered objects and views

The GeoMatrix™ Toolkit reduces time-to-market and R&D costs by providing:

- proven technology
- a high-level object-oriented API
- high performance global rendering services
- example applications

The GeoMatrix SDK consists of six C++ modules (Figure 1), each providing a set of objects and services that are necessary in a digital Earth application. The *gmx* module is the heart of the system and provides object management and rendering services. The *gmx_dlib*, or data library, handles the reading and writing of dataset tiles from local disks or remote servers. Interaction and view control are handled with the *gmx_view* library. Fonts are provided by the *font* library. The pre-processing of input data sources are handled by the extensible *gmx_load* library, used for reading in source data in various formats, and the *gmx_tgen*, or tile generator library, generates a multi-resolution tiled representation of the source data that conforms to the GeoMatrix global gridding system.

The GeoMatrix system dynamically creates the requested view of the Earth by assembling only the required tiles of data, rather than maintaining a static model of the Earth. Therefore, it is maximally efficient, and it is also flexible, since the tiles can be drawn from distributed storage locations.

Tiles are the basic building blocks of a GeoMatrix digital Earth. The Tile Generator (*gmx_tgen*) module is used to derive tiles from a source, such as a TIF image file. Tiles of data are typically stored for high-speed access at rendering time. The software architecture of the GeoMatrix system is a hierarchy of C++ objects starting with one or more *source* data objects, and *tile* objects related to those sources.

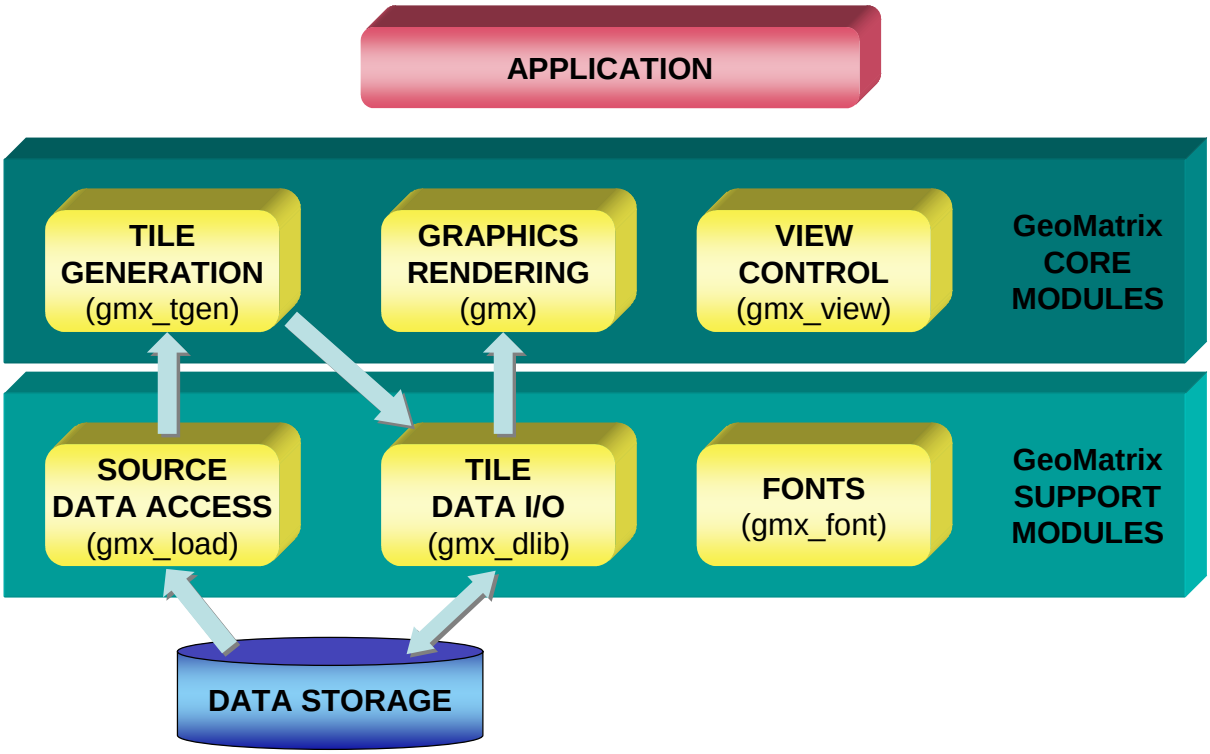

Figure 1: GeoMatrix modular architecture

Tiles are then associated with a *dataset* object, which treats them as an identifiable set with coherent geographical extent, consistent resolution, data format, etc. Multiple *datasets* can be viewed together by being attached to the same globe object (Figure 2). A globe object is essentially a collection of *datasets*, and is spherical in the sense that the *tiles* in the attached *datasets* all conform to a particular tiling method. The grid of *tiles* that covers the global sphere is called the "geomatrix".

As a collection of *datasets*, globes can only be viewed when attached to a *scene* object. A *scene* stores information about a graphics window and draws a procedurally generated sphere for each *globe* attached to it. A *scene* will typically have a single *globe*, but it can have multiple *globes* that are coincident, concentric or offset. An application can then, for example, blend between two *globes* that have different *datasets*. It can show a *globe* of cloud data at some altitude over a *globe* with terrain data. It can show a planet with its moon revolving around it, and can show multiple windows with different perspective views of the same *globe*. Thus, with the object hierarchy of *sources*, *tiles*, *datasets*, *globes*, and *scenes*, the application's developer has tremendous flexibility and power to create the right solution.

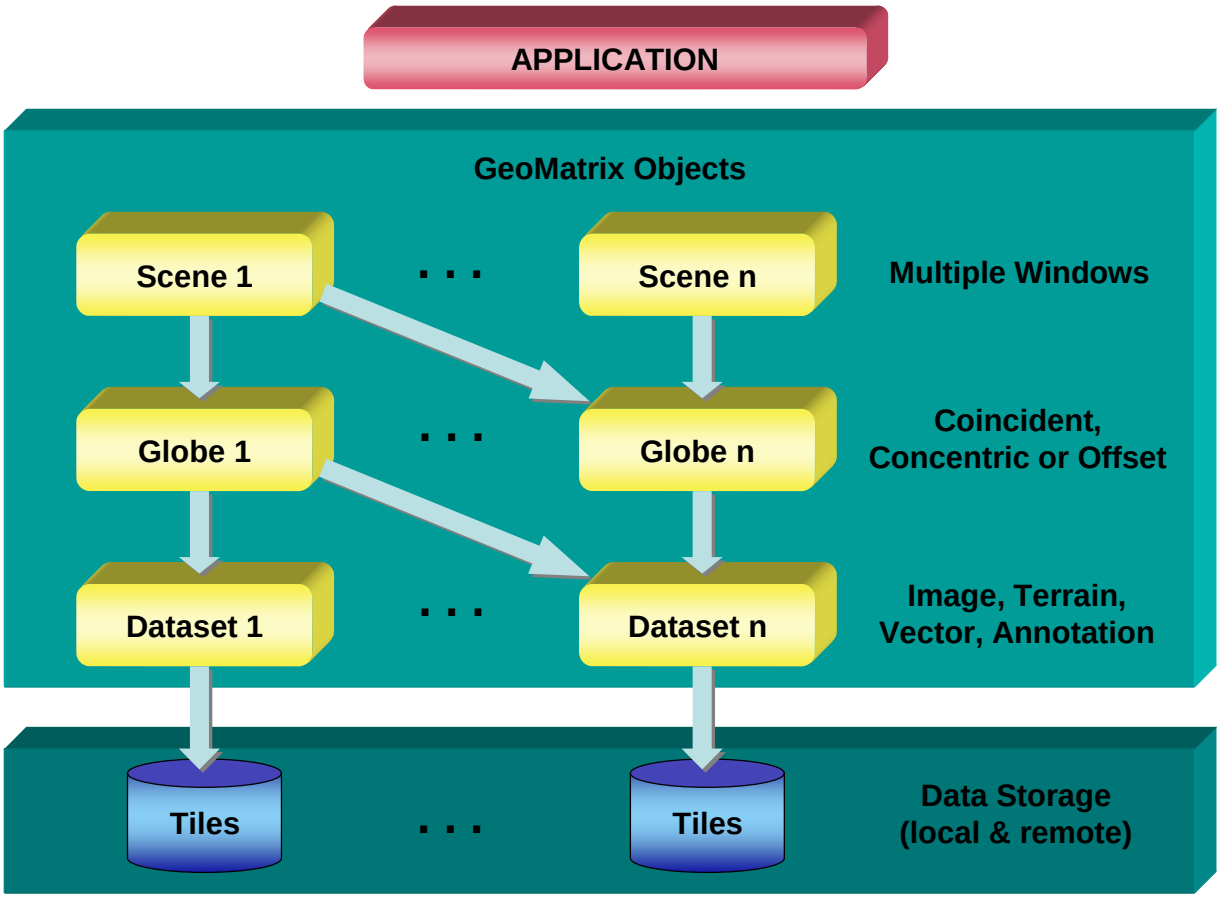

Figure 2: GeoMatrix object hierarchy

The GeoMatrix system does not control your graphics environment. It does not have a scene graph. Paradoxically, although it performs sophisticated, state-of-the-art operations that can handle terabytes of data and render breath-taking views of the Earth, from a developer's standpoint it is just another node in your scene graph. This gives the developer maximum power to control the result. Although it is just a node in your scene graph, the GeoMatrix system provides a wide variety of utilities, to query the scene object and get the information you need to position and successfully draw any objects (lines, text, 3-D models) you want in relation to the digital earth. The *gmx_view* module also provides sophisticated methods for controlling views of the digital earth, which can save dozens and perhaps hundreds of hours of effort.

GeoMatrix Toolkit Features

Data:

- Multi-resolution support for four basic data types: imagery, terrain, vector, and annotation.
- Datasets are independent. Ability to attach or detach, and render an arbitrary number of datasets each with an arbitrary, potentially very large, amount of data.
- Supports any number of overlapping, high-resolution **image or terrain** data inserts.
- Datasets can be accessed locally or remotely via HTTP. (Data loader is extensible for loading data using other protocols).
- Unique sphere tiling method enables the whole earth, including polar areas, to be efficiently represented and rendered.
- Supports industry standard texture compression formats.
- Lossless compression of datasets further reduces disk and bandwidth requirements

Rendering:

- Rendering and display of **global and local** scale imagery, elevation, and vector-based datasets.
- Rendering of **multiple** global and high resolution imagery and terrain data inserts.
- Continuous level of detail while zooming.
- Terrain morphing between resolution levels.
- Terrain scaling and tessellation bias control.
- Flat or spherical globe rendering.
- Ultra-fast Level-Of-Detail calculation.
- Demand paging of tiled data for four basic data types: imagery, terrain, vector, annotation. Memory resources are efficiently recycled among the currently used memory pages.
- Object interface and rendering engine allow for multiple scenes (windows), each with multiple globes, and each globe with multiple datasets.
- Multiple globes can be coincident, concentric, or separated to provide various effects such as translucent overlays (used with clouds above the Earth), fade between datasets, geologic features, planets, etc.
- Scene scaling allows both global and street level features to be rendered while retaining numeric precision.
- Ability to add your own polygonal model for airborne, ground, and space-based objects.
- GeoBrowser application builders can add support for other types of data to be drawn onto the world. This can include points, lines, images, 3D models, etc.

Interaction:

- Interactive, pan, zoom, tilt, and spin control.
- Interactive movement via Earth pivot mode or flight mode.
- Three ways of managing perspective: matrixes, “view parameters”, quaternions.
- Utility functions for culling and picking objects in a scene, facilitating coordination of cultural features, and collision detection.
- Application programmers can add their own content as well as look and feel to GeoMatrix-based digital earth applications.
- Support for screen dumps at any resolution to popular image formats.

