

## **3D-MODELING TECHNOLOGY AND METHODOLOGY AND VIRTUAL PHILADELPHIA DEMO**

VICTOR SHENKAR  
Director and Responsible for the Researcher  
GeoSim Systems LTD  
Petach-Tikva - Israel  
[www.geosimcities.com](http://www.geosimcities.com)

---

### **1. Background**

1.1 The terms “virtual city”, “cyber town” or “digital city” frequently referred to in computer applications, are being employed for a variety of information contents and interfaces, exhibiting different levels of quality and functionality.

1.2 The existing “cyber-copies” of actual cities do not convey to their users genuine sense and perception of “real” urban places.

Most of the current activities aimed at creating “true” virtual cities with photo-realistic built, richness and fullness of geo-referenced information content, crucial to support professional users and business applications, social interaction and digital entertainment are still at their early research stage.

1.3 Currently practiced methodologies to produce precise and complete 3D-city models are extremely laborious and time-consuming.

A major improvement of productivity in data collection and data processing is needed to transform the existing modeling tools into an industrially viable solution for high-fidelity, large-scale city modeling.

1.4 Another major obstacle to efficient creation and exploitation of city model based applications is the lack of standards and incompatibility of the existing data collection, data processing and visualization tools.

To overcome this obstacle a system-level integration of such tools is required.

### **2. General Description of GeoSim Modeling Technology and Methodology**

#### **2.1 Block Diagram**

The diagram in Fig.1 below graphically depicts the GeoSim city modeling technology and methodology, presenting its overall functionality as well as the general flow of city model production activities, as further explained below.

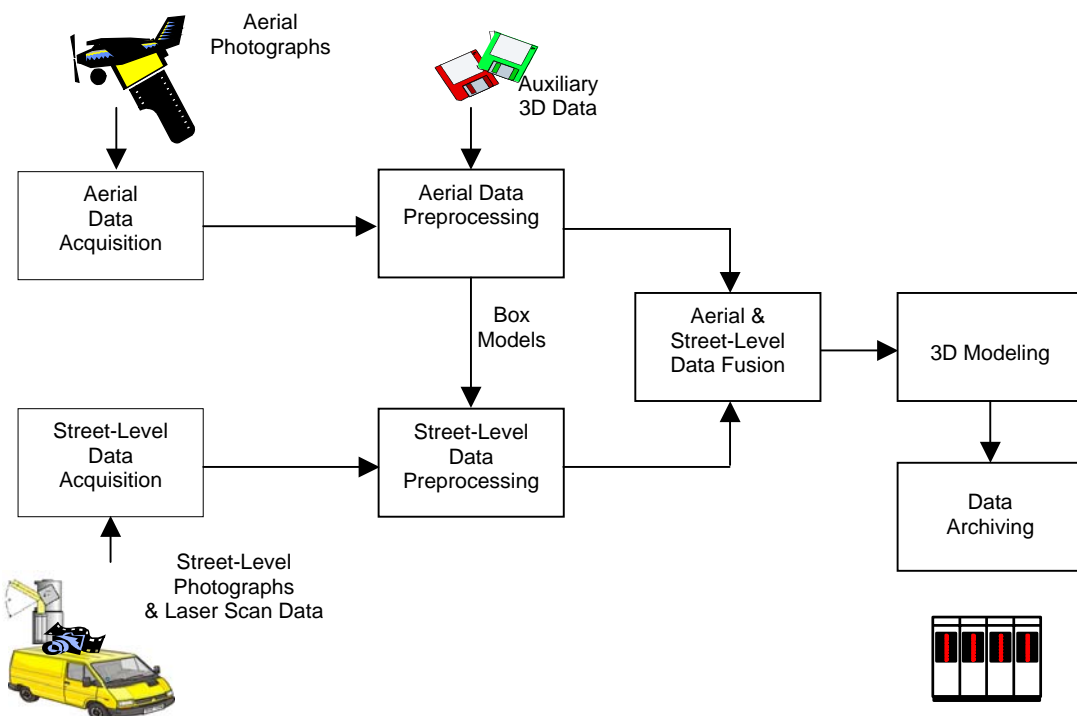


Fig. 1: GeoSim 3D-Modeling Technology and Methodology

## 2.2 Aerial Data Acquisition

Aerial data acquisition involves collection and import of two sets of overlapping air photographs. Such images are standard aerial photography products, based on 150mm and 300mm focal-length mapping cameras.

To provide adequate coverage of the modeled part of a city, the photography missions are flown along carefully pre-planned flight trajectories, and taken at both vertical and oblique angles.

“Hard-copy” negatives produced by the above cameras are scanned with precise photogrammetric scanners in order to convert them to corresponding digital files.

## 2.3 Aerial Data Pre-Processing

The following two rather standard pre-processing functions and one more specific, application-oriented function are applied to imported aerial photographs:

Image solving (based on Ground Control Points – GCP's), ortho-mosaicing and orthomosaic-based segmentation of the modeled part of a city into "city blocks".

Stereo extraction of 3D polylines outlining all the buildings within each city block and city block borderlines.

Proper alignment, rectification and binding of the aforementioned 3D polylines into spatially corrected “modeling units” (or "box models") representing individual buildings within a "city block".

Due to high-resolution airborne imagery and high-precision GCP's, the resulting box models have spatial accuracy of 4"-8" (location and elevation-wise).

Such box models serve as very precise geodetic foundation for the rest of the 3D-modeling process.

## **2.4 Street-Level Data Acquisition**

Street-level data acquisition involves collection and import of street-level photographs and laser scans needed for reconstruction of details indiscernible from aerial photographs.

The street-level data is collected by a ground vehicle carrying an integrated set of sensors and electronic gear, which include:

Two digital (still) cameras, producing lower elevation and upper elevation images.

A two-axis laser scanner, providing fast, dense and precise range measurements ("point clouds").

Attitude and navigation subsystem, consisting of a differential GPS, attitude sensors and a wheel-attached rotary encoder.

A PC with special interface cards, serving as system controller, communication driver and intermediate storage media.

This instrumented ground vehicle, also referred to as Street-Level Data Collector (SLDC), collects the street-level data while moving and stopping along pre-planned driving routes. Such routes have to pass along all the streets of the modeled part of a city.

### **Street-Level Data Pre-Processing**

The following main pre-processing functions are applied to imported street-level data:  
Alignment and calibration of all sensors installed in SLDC.

Initial registration, representing coarse location and orientation of SLDC at the image/scan taking instances (i.e. at stop points).

### **Assignment of street-level data to "city blocks"**

Another useful street-level data pre-processing function combines up to six different still images taken at one stop point into one unified panoramic image. Such panorama, which significantly enlarges the instantaneous field-of-view of digital cameras, reduces the number of needed stop points and greatly facilitates recognition and 3D-modeling of street-level details.

### **Aerial and Street-Level Data Fusion**

To efficiently use both aerial and street-level data in the ensuing 3D-modeling process, these two data sets have to precisely merge into one unified data set.

Such data fusion is carried out in a two-step process:

The "precise relative solution" stage, in which the initial registration results are reprocessed to obtain more precise relative location and orientation of SLDC at all stop points.

The "precise collective solution" stage, which adjusts the precise relative solution mentioned above as to achieve the best fit between street-level data and corresponding aerial photos and box models.

In short, once the precise collective solution has been established, aerial and street-level data become fully co-aligned and can be freely intermixed with each other during the course of 3D-modeling process.

## **3D-Modeling**

3D-modeling is the most crucial and time-consuming activity in the entire city model production process.

Given the current state-of-the-art, precise, high-fidelity 3D-modeling of urban environments has still to rely on visual perception and manual skills of human operators ("3D-modelers").

To achieve a significant productivity improvement, 3D modelers have to be provided with a user-friendly, fully-interactive 3D-modeling environment as further described below.

GeoSim's core 3D-modeling environment is the Autodesk 3D Studio VIZ (one of most popular and most extensive 3D-modeling solutions) augmented with a number of seamlessly integrated proprietary plug-ins.

Applied in conjunction with the standard 3D Studio tools and procedures, these plug-ins have a dramatic impact on improved efficiency of the 3D-modeling process.

A complete 3D-modeling cycle is executed on a city-block by city-block basis; i.e. each 3D-modeler is assigned with a task of producing a complete 3D city-block model, carried out in the following main consecutive phases:

### **Development of Building Models (BM)**

The BM are a digital outdoor representation of houses and other man-built structures ("buildings") located within city block boundaries, by means of a two-part data structure: side-wall/roof-top geometry and side wall/roof top texture (RGB colors).

Side-wall and roof-top geometry is represented by a specific polygonal mesh. 3D-coordinates of mesh vertices are geo-referenced and have absolute location (i.e. location with respect to a geographic datum/projection system being used as an underlying 3D-coordinate system) accuracy of 4"-8". Side-wall/roof-top textures are mapped from the "texture library" extracted from the aerial and street-level photographs and accumulated during the course of 3D-modeling process, as to represent in a concise and reasonably realistic manner the actual textures. Such mapped textures have resolution better than 1"/pixel.

### **Development of Terrain Skin Model (TSM)**

The TSM represents paved and unpaved surfaces of the terrain skin located within city block boundaries by means of a two-part data structure: surface geometry and surface textures (RGB colors).

Terrain skin surface geometry is represented by a specific polygonal mesh. 3D-coordinates of mesh vertices are geo-referenced and have absolute location accuracy of 4"-8". Terrain skin surface texture is mapped from the "texture library", as to represent in a concise and reasonably realistic manner the actual textures. Such mapped textures have resolution better than 1"/pixel.

### **Development of Street-Level Culture Model (SCM)**

The SCM represents "standard" urban landscape elements located within city block boundaries, including illumination poles, traffic lights, traffic signs, trash cans, bus stops, benches, trees and vegetation, etc., by means of a two part data structure: object surface geometry and object surface textures (RGB colors). The object surface geometry and textures of street-level culture are mapped from the "street-level culture library" as to represent in a concise and reasonably realistic manner the actual street-level culture. The street level library objects are "implanted" into the TSM within 4"-8" absolute location accuracy (as compared to location of actual street-level culture).

### **Creation of Levels-of-Detail**

In addition to partitioning of a large-scale city model into city-blocks (to enable so-called online "data paging"), efficient real-time image rendering of such city models also requires that all three data layers presented above possess a special data structure referred to as levels-of-detail (LOD).

What the LOD mechanism practically means is that all 3D-objects composing the BM, TSM and SCM have 3 different representations, typically descending in their file sizes by a 5-10 ratio from one LOD to another.

## **Data Archiving**

The data archiving function is built upon an enterprise-class, object-oriented database with spatial search capabilities. It enables gigabytes of image data and 3D vector data in the 3D Studio MAX format to be stored and managed under a version control.

Coupled with the 3D modeling tools, this database facilitates simultaneous work of multiple modeling teams while building and updating 3D city models, as well as during their routine maintenance phase.

## **2.9 Implementation of GeoSim 3D-Modeling Environment**

All hardware components of GeoSim city modeling environment are commercially available off-the-shelf items.

Computing hardware is based on standard Pentium III/IV PC's in three basic configurations:

Industrial PC with special-purpose I/O cards supporting SLDC data collection and storage.

Windows 2000 Workstations supporting operators dealing with data pre-processing, data fusion and 3D-modeling.

Windows 2000 Servers supporting data storage, archiving and client-server communications within the city model production facility.

Software components combine both: commercial off-the-shelf products (primarily Microsoft Windows, Autodesk 3D Studio and Erdas Stereo Analyst) and GeoSim's special-purpose applications, developed under Microsoft Developer Studio/C++. All components are tightly integrated into one unified client-server environment, which enables unprecedented productivity and streamlining of the city modeling process.

Being a collection of standard "building-blocks" (commercial, off-the-shelf hardware and software components with standard, client-server interfaces), this environment can be implemented and deployed in an extremely scalable manner, which is a pre-condition for effective production of large-scale, high-fidelity city models.

## **3. General Description of GeoSim Browsing and Visualization Environment**

### **3.1 Block Diagram**

The diagram in Fig.2 graphically depicts the GeoSim browsing and visualization environment, showing its general client-server architecture, as well as the flow of data and user's commands.

This environment allows for browsing and visualization of GeoSim city models in real-time, supporting at the same time a variety of city model applications, as further explained below.

### **3.2 Databases**

GeoSim's client-server browsing and visualization software processes and manipulates in real-time two main databases:

#### **3DM**

3DM is a 3D-city model converted from the 3D Studio VIZ as described in Section 2.7 above, into a proprietary format needed for efficient, real-time image rendering. The 3DM is delivered to the users on a CD/DVD, or downloaded via the Internet off-line (prior to an online session).

During the online session the 3DM has to reside locally at the client's side.

## “Other Contents”

“Other contents” consist of three main data layers:

A collection of geo-referenced links between selected locations within the 3DM to a collection of geo-coded Web pages and Web sites.

Geo-coded Web pages mentioned above (comprising text, images, video and audio), represent city-related demographic, commercial, engineering, cultural and other data.

IDL data represents personal/institutional user identification and user’s current virtual location (“present position”) within the 3DM.

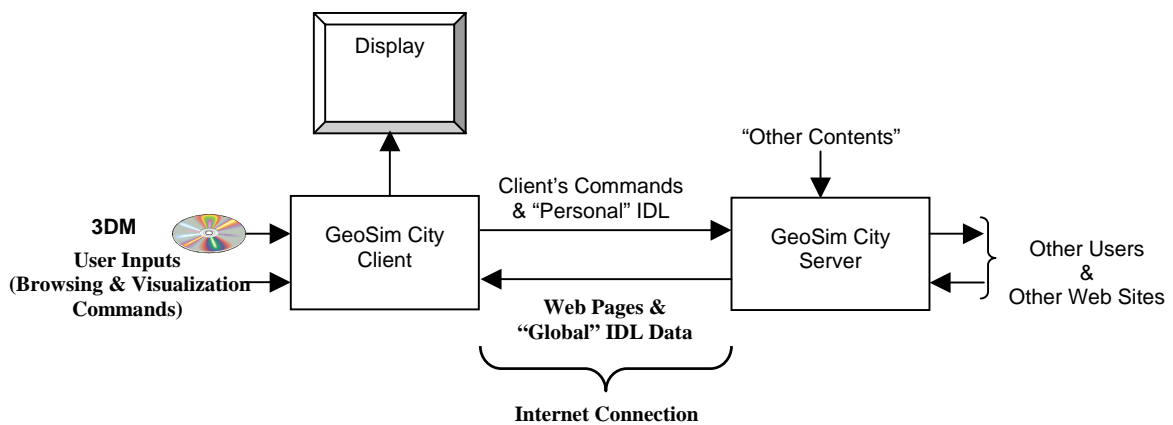


Fig.2: GeoSim City Browsing and Visualization Environment

During the online session, all “other contents” are streamed in real-time over the Internet.

The above data partition between the locally and remotely residing data is a well-balanced solution, which minimizes data streaming requirements due to the following complementing features of partitioned data:

3DM's are huge data files, not amenable to real-time transmission over the Internet, and they represent content, which does not need frequent updates.

“Other contents” are much smaller data sets, but they require very frequent changes and modifications.

## Operational Functions

Running under stringent, real-time constraints, GeoSim's browsing and visualization software applies to the 3DM and “other contents” a wide range of operational functions, such as: client-server communications, 3D navigation, 3D pointing, IDL tracking, etc.

But the most demanding function is image rendering and animation. This function generates perspective views of the 3DM and of other contents as “seen” from the user's point-of-view (POV) and line-of-sight (LOS).

To be able to generate such images with a sufficient frame-rate on a commercial grade hardware, an intricate string of sequential and parallel data/image processing functions have to be put together, optimizing the data format accordingly.

The most critical data/image processing functions are as follows:

Data paging and occlusion culling – for any given view port (POV/LOS), this function determines the minimal subset of the 3DM, which is visible from the user's POV/LOS (excluding hidden surfaces).

For urban landscapes, in which a close building may obscure all other buildings along the LOS, this function may produce a dramatic reduction in the polygon count.

Level-of-Detail - for a "post-occlusion-culling" 3DM subset, this function determines (based on the viewing distance) the optimal LOD for all 3D objects included in this subset.

The LOD function also contributes a major reduction in the polygon count (avoiding rendering of excessive levels-of-detail).

Image rendering – although this function is a standard graphic library function (DirectX), it has to be implemented with a great care so that standard graphic cards ("accelerators") performing this function are able to operate in a most efficient manner.

In particular, the download of 3DM subsets from the hard disc to the graphic accelerator has to be done with minimum or no data conversion, and image rendering has to be organized in such a way that large amounts of geometry are rendered by a single command with a single texture setting (i.e. applying "optimal" texture grouping).

## **Implementation**

The client-side hardware is based on a standard PC (Pentium III/IV) with a commercial-grade 3D graphic card.

The client-side software is based on the Windows OS and DirectX library, bundled with special-purpose utilities developed under MS-Visual Studio.

The server-side is hosted on commercially-available Web servers, running GeoSim's special-purpose application built upon the SQL Server.

## **4. Summary**

4.1 3D-modeling technology and methodology presented in Section 2 combine a number of commercial off-the-shelf products with GeoSim's proprietary hardware and software tools.

Built into an integrated, multi-user environment these tools provide unprecedented productivity in high-fidelity, large-scale city modeling.

The 3D-browsing and visualization solution presented in Section 3 is optimized for efficient handling of high-fidelity, large-scale city models. This solution supports a wide range of Web-enabled, city-model-based applications while running on standard, commercial-grade PC platforms.

Finally, Fig.3 and 4 below show two examples of urban scenes rendered from GeoSim city models.



Fig. 3: Virtual Milan (Piazza La Scala)



Fig. 4: Virtual Philadelphia (City Hall)