HANDLING LARGE TERRAIN DATA IN GIS

Wanning Peng, Dragan Petrovic, Clayton Crawford

ESRI
380 New York St., Redlands, CA 92373, USA
wpeng@esri.com, dpetrovic@esri.com, ccrawford@esri.com

KEY WORDS: Database, Large, DEM/DTM, Multiresolution, Generalization, LIDAR, GIS

ABSTRACT

This paper presents a research and development project that will provide an extension to 2D geo-databases for handling large terrain data. It first discusses application requirements and system design, and then elaborates system architecture for optimal data organization and updating, efficient multi-resolution queries, and dynamic DTM generation. It then addresses technical issues related to data storage, seamless tiling, vertical indexing, and DTM generalization. Finally, it discusses the limitations and shortcomings of the current approach, and identifies future research and development tasks.

1. INTRODUCTION

Many GIS projects, especially statewide and nationwide ones, often need to store and manage large terrain data. Even small-scale projects may have to deal with a large amount of terrain data, due to newly available data acquisition techniques such as LiDAR. Such data can be several tera-bytes in size, or may contain billions of measurement points.

While most of today's enterprise geo-databases (such as SDE) are capable of handling large 2D data, terrain data have brought new requirements and challenges. These include 1) how to integrate terrain data with 2D data, 2) what data structure to use, and 3) how to support high performance multi-resolution spatial queries and update.

Given the fact that TIN and GRID are the most popular data formats in digital terrain modeling, it is necessary to examine if they are the best choices for storing terrain data. Because different applications may require data of different spatial resolutions depending on underlying conceptual models (Peng, 2000, 1997), multi-resolution queries are becoming a more and more important subject in GIS. Some applications may even require a so-called "horizontal" multi-resolution query that specifies different levels of vertical resolutions for different parts of a study area (Kinder et al., 2000). Typical examples include landscape planning and 3D flight simulation, where the center of interest often requires higher resolution data, while the rest of the area only requires data of coarser resolutions.

To address all these issues, and others, a new research and development project has been implemented at ESRI to provide an extension to current 2D geo-databases for handling large terrain data. The rest of the paper elaborates the design concept and system architecture, and addresses related technical issues. Finally, it provides an outline for further research and development.

2. DESIGN CONSIDERATIONS AND SYSTEM ARCHITECTURE

The design can be boiled down to three aspects: 1) *what* to store; 2) *where* to store it; and 3) *how* to store it.

2.1 What to Store

Typically, source terrain data include 1) measurement points (e.g., spot height points such as LiDAR data), 2) contours, and 3) structure lines (or break lines) that capture the discontinuity of terrain and other important geomorphologic and geographic features. Because a collection of individual points, contours, and break lines, does not constitute a good (continuous) terrain representation in a digital environment (Peng et al., 1996), they are not usually directly used for surface visualization and analysis in GIS. Instead, a typical GIS would build a digital terrain model (DTM) using these data, and carry out analysis based on the DTM. Because of this, people often store and manipulate their terrain information directly as a DTM, disregarding the source data.

A DTM may take the form of a GRID or TIN (triangulated irregular network). Spatial resolution of a GRID DTM is inherently constrained to cell size – the smaller the cell size, the higher the resolution – apart from the quality of the original data. However, once generated, the source data are lost and no improvement is possible. One can only down-sample a GRID DTM (i.e., go to a larger cell size and, thus, lower resolution). Creating a new DTM of a smaller cell size out of an existing GRID DTM will not increase its spatial resolution. A TIN DTM, on the other hand, does not suffer from this constraint due to its adaptive nature, although a small elevation tolerance may be employed to reduce data quantity in constructing a TIN.

Many large data providers (USGS, for instance) choose GRID for their terrain data, due to its simplicity and relatively small storage size. TIN is typically used in places where engineering precision is required. Because of its sophisticated structure and heavy overhead in storage (in order to keep topology), TIN is rarely used to provide and maintain a large amount of terrain data.

Obviously, GRID is preferred if format simplicity and storage space are the concerns. However, TIN might be a better choice if high precision is desirable, especially when terrain skeleton information (such as break lines and local extreme points), and other structure lines are important to preserve. A hybrid system that uses both GRID and TIN may sound like a good solution, if only it does not increase the complexity and difficulty in data management and updating, as well as in determining when to use GRID and when to use TIN.

Storing source data may seem, at the first glance, unacceptable, as source terrain data are not suitable for surface visualization and analysis in GIS. Close examination, however, has led to a new conclusion. *First*, both TIN and GRID DTMs are results derived from source terrain data by applying certain topological and spatial rules. This is done due to their advantage in computer analysis and terrain relief visualization, not because they can provide better data quality, or are easier to manage and update. In fact, if source terrain data are properly arranged and stored (to be discussed in the next two sections), a GIS can always generate a DTM using these data dynamically, with high performance. This will allow users to take the advantage of both TIN and GRID without being burdened by TIN's storage overhead or GRID's precision problem.

Second, source terrain data often have many uses. For instance, road networks provide critical data for transportation and planning applications. They also play an important role in DTM construction. Drainage systems are another typical example. Even with mass point data (such as LiDAR), there may be attributes associated with each point that may serve other applications (e.g., vegetation analysis). Storing terrain data separately as a DTM would mean an increase in storage space, extra work in data management and updating, and extra complexity and difficulty in keeping data synchronized. In other words, storing source data would allow the same data to be shared by different applications, minimize storage space, and eliminate or reduce the work in data synchronization.

Third, when it comes to multi-resolution queries, source vector terrain data allow structure lines to be generalized according to a user specified target resolution (Weibel, 1992; Peng et al., 1996). Whereas in a GRID DTM the original skeleton information may have already been distorted or lost. In a TIN DTM, although skeleton information may be preserved as constraints in the triangulation, attributes associated with each structure line still need to be handled. Kidner and colleagues (Kidner et al., 2000) also provide some good arguments against storing (explicit) TINs.

Finally, storing source data is flexible. One can always modify or redefine the rules, change source data and their combination, and create different DTMs accordingly, which is useful for research projects. Trying to find a better triangulation criterion for TIN DTM construction, or a better interpolation method for GRID DTM generation, are good examples.

2.2 Where to Store

GIS applications often require a database environment that supports (among others) 1) geometric and thematic description of spatial objects, 2) topological relationships at a geometric primitive level and object level, 3) versioning, 4) multi-user access and editing, and 5) seamless, scalable, multi-resolution, and high performance spatial queries.

The approach described is based on ESRI's geo-database framework defined in ArcGIS (Zeiler, 1999), as it has the potential to meet the requirements listed above. As shown in Figure 1, a *geo-database* (or database for short) contains one or more *feature datasets*; a feature dataset contains one or more *feature classes*; a feature class contains one or more *features* of the same *geometric type* (point, line, or area). A feature dataset defines a conceptual entity for those feature classes that share the same spatial reference, cover the same geographic extent, and are often thematically related to each other.

Under this framework, it is clear that source terrain data should be grouped into different feature classes, according to their geometric type, source, and thematic description. Typical examples include drainage systems, mass points (e.g., LiDAR data), road networks, ridgelines, and ground control points. Those feature classes that contribute to the same ground area are then put into the same feature dataset.

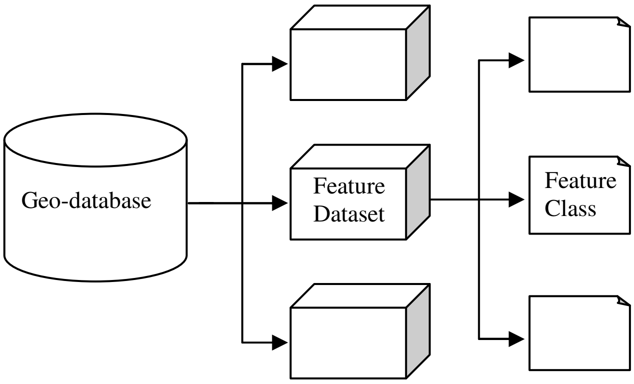


Figure 1: A geo-database structure.

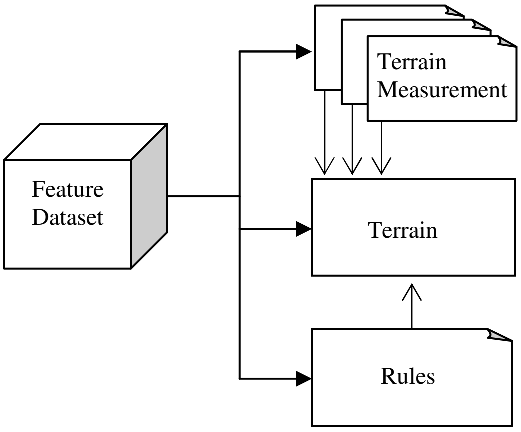


Figure 2: Terrain in geo-database.

2.3 How to Store

Feature classes of terrain data do not themselves have much meaning in terrain application. A higher level of abstraction is necessary in order to support users to properly model their terrain data and applications. This is achieved by introducing *Terrain* into the geo-database framework described in section 2.2. A Terrain is defined as a special type living inside a feature dataset, consisting of one or more feature classes within the dataset and a set of *rules* (Figure 2). Feature classes constituting a terrain dataset are called *terrain measurements*.

Terrain measurements can be stored as point, line, or area feature classes. They may participate in more than one Terrain within the same feature dataset. Terrain measurements are also regular feature classes that can be used for other purposes, allowing 2D data and terrain data to be integrated into the same database.

The set of rules defines how the measurements are applied when constructing a terrain representation (such as a DTM) as a result of spatial query. These rules 1) define the role of each measurement, whether a measurement should be added as a mass elevation point, a break line, a replacement polygon, or a clipping polygon, and so forth; 2) specify, for each measurement, where the elevation information comes from, whether the Z coordinate, the value of a particular attribute, or a result interpolated from a given DTM (which may include the one under construction); and 3) specify at what resolution, or resolution range, a measurement should participate. This is necessary in order to support multi-resolution queries and accommodate different requirements in terrain generalization (to be discussed in section 3.2).

The introduction of Terrain allows a geo-database to store and manage terrain data without being bound to a particular type of DTM. A DTM can be generated dynamically upon users' request by applying the rules. The idea of storing measurement data (rather than explicit DTMs) in a database is similar to the philosophy of *Implicit TIN* (Kidner et al., 2000).

2.4 The Basic Requirements

Terrain provides a meaningful and comprehensive entity through which users manage, query, and apply their terrain relief information. Such an entity is referred to in ArcGIS as *terrain dataset*.

In order to support various applications, a number of basic requirements have been identified that Terrain should support. It is also assumed that a DTM is still the most favorable structure for surface visualization and analysis in GIS. Therefore, a spatial query on a Terrain is expected to result in a DTM, upon which various analyses can be performed. These requirements include:

- Support a large area extent and a large amount of data
- Support point, line, and area data
- Support update, on both measurements and rules
- Allow certain measurements to be included/excluded in a spatial query
- Support TIN and GRID DTM output
- Support spatial query with respect to a given area of interest and vertical resolution. This would require Terrain to dynamically generate DTMs of given resolutions, anywhere within the extent of the Terrain
- Support "horizontal" multi-resolution query – a special kind of query that specifies different vertical resolutions for different parts of a given area. The query will result in a multi-resolution DTM in which vertical resolution varies across the whole area

3. KEY TECHNICAL ISSUES IN SYSTEM DEVELOPMENT

To implement the supports listed in section 2.4, three key technical issues need to be addressed: 1) tiling, 2) vertical indexing and DTM generalization, and 3) data updating.

Although today's good geo-databases are capable of handling large amounts of data, and 2D spatial indexing is basically a built-in feature, extra arrangements are still necessary in order to support fast DTM creation, multi-resolution queries, and DTM generalization. These include internally arranging measurement data into *tiles* according to data extent, density, and hardware/software constraints; and introducing extra indexing in the *vertical* dimension.

3.1 Tiling

Because data are potentially huge, it is not feasible to handle all data at once. Not only is memory a problem, but performance can be unacceptably poor. Tiling, on the other hand, can be a good "divide and conquer" approach for handling data of large extent, given the constraints of today's available technology. A good tiling scheme can result in spatial coherence – data are organized and stored according to their spatial proximity, thus increasing the performance in spatial query and data transfer. It also provides a powerful (2D indexing) mechanism for fast searching; allows data to be handled in a more manageable form; allows memory and CPU intensive tasks to be performed locally without paralyzing the system; and is essential for DTM generalization (to be discussed later).

Tiling divides a large geographic area into smaller, more manageable, units (Figure 3), which can have different forms. This approach uses a regular rectangle tile for its simplicity and efficiency in computation. Choosing a proper tile size is a bit more complicated. It depends on data density, CPU speed, available memory, and other considerations. Basically, the size must not be too big, so that a full resolution DTM of any tile can be generated using an acceptable amount of system resources.

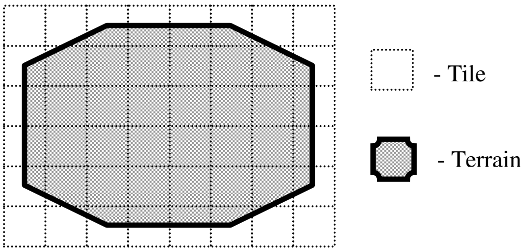


Figure 3: Divide large terrain extent into regular tiles.

Tiles are used as the basis for reorganizing points. Points falling into the same tile can be grouped into, and stored and handled as, one single entity – a so called *multi-point*. This will reduce storage space, increase spatial coherence and access speed, and reduce disk I/O and network traffic. As most of the terrain data will be mass points coming from remote sensing and laser scanning, the benefit of this process can be significant. Line and area features may require extra work in order to benefit from this measure, as a single feature can cross many tiles.

3.2 Vertical Indexing and DTM Generalization

The vertical indexing is introduced to quickly identify those data that contribute to a certain given resolution, so that no redundant data will be retrieved and used to generate the output DTM. This is achieved by assigning points to different "layers" according to certain rules – a preprocessing step executed when a terrain dataset is created.

Vertical indexing identifies points, and vertices of line and area features, that contribute to a DTM of a specific vertical resolution, or a specific layer in a DTM Pyramid (Floriani, 1995). A DTM Pyramid is composed of a list of pyramid layers, with the first layer corresponding to the full resolution DTM, and the last layer corresponding to the DTM of least resolution. Therefore, vertical indexing can be seen as sorting data according to the pyramid layer position (index).

The vertical resolution for each layer is relative to the full resolution DTM. The number of layers in a pyramid, and each layer's (relative) resolution, are up to the user to define. Generally speaking, the more the layers, the smoother the transition between these layers. Unlike image pyramid layers, increasing the number of layers in a Terrain pyramid will not result in more data to be created, duplicated, and stored. It merely increases the number of classifications. However, it does increase the preprocessing time, and potentially the number of multi-points when points within the same tile are further divided into subgroups based on their vertical indices (to be discussed later).

Pyramid layers can be built by deriving a DTM of lower resolution from the full resolution one, through generalization (Weibel, 1992; Peng et al., 1996). A number of algorithms have been published in the literature, such as DTM filtering (Loon, 1978; Zoraster et al., 1984), DTM compression (Gottshalk, 1972; Heller, 1990), and structure or skeleton line generalization (Wu, 1981; Yoeli, 1990; Wolf, 1988; Weibel, 1989). An evaluation of these three types of methods can be found in (Weibel, 1992). Other algorithms are also available in the area of computer graphics, mainly to serve real time visualization (Kalvin, 1996; Hoppe, 1998; Lee, 1998; Reinhard, 1998). This design adopts the DTM compression (or point decimation) approach for point features.

Line and area features require a generalization approach that takes into account topological relationships and the vertical dimension. Unfortunately, there is still no good algorithm available for automated generalization of line and area features. Furthermore, different applications may have different generalization requirements and criteria. Based on these considerations, this design introduces three mechanisms to index line and area features: 1) use user provided multiple versions of pre-generalized terrain measurements, and associate each version with a corresponding layer in a DTM pyramid; 2) adopts Line Generalization Tree (Johns and Abraham, 1987), but supports more algorithms; 3) uses on-the-fly automated generalization of the original measurements.

The Line Generalization Tree has a limitation that only selection of vertices can be performed. This project will focus next on developing algorithms for on-the-fly automated generalization, and the enhancement of the Line Generalization Tree.

Vertical indexing adds another control for grouping data within the same tile. Instead of putting all the points within the same tile into one group, only those points that share the same vertical index will be grouped into a single multi-point. Because points are organized according to their corresponding tiles and vertical indices, spatial queries can retrieve data efficiently.

3.3 Updating Data

Requirements for terrain update come from two aspects: the measurements, and the rules. Any changes regarding these two will require the internal vertical indexing to be updated. Because rules are private to the terrain dataset, updating rules is simple and straightforward. Measurements, on the other hand, are shared by other applications, and can be modified without going through terrain datasets. In order to keep terrain datasets and measurements in sync, some mechanisms are required that keep the datasets informed whenever an update is performed on the measurements. This is done through *Events* and *Invalidated-Area*. An Invalidated-Area is a region where changes of measurements have occurred. It allows an outdated terrain dataset to be updated locally.

When an update to a measurement is committed, an Event is broadcast. Those terrain datasets that are affected will update their Invalidated-Areas upon receiving the Event. Users will then decide when to update the affected terrain dataset.

4. APPLICATION EXAMPLES – SPATIAL QUERY AND SURFACE ANALYSIS

Spatial query and surface analysis are Terrain's two most important applications. A typical spatial query takes an area of interest *AOI* and a (relative) vertical resolution ΔH, and outputs a TIN or GRID DTM (specified by the user, Figure 4). The output can be an (transient) object that will be persisted only if requested by the user. Area of interest *AOI* may contain multiple regions. In this case, there will be a list of ΔHs, each of which corresponds to a region in *AOI*. A multi-region *AOI* will result in a multi-resolution (continuous) DTM, while a single-region *AOI* will produce a single-resolution one. With all the indexing support, the system can quickly allocate those multi-points that contribute to ΔH but are also within the query area *AOI*. Line and area measurements can also be quickly identified using vertical indices.

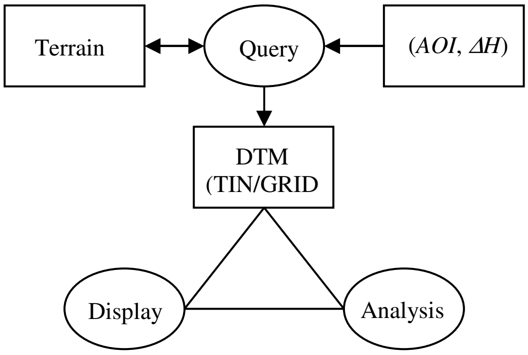
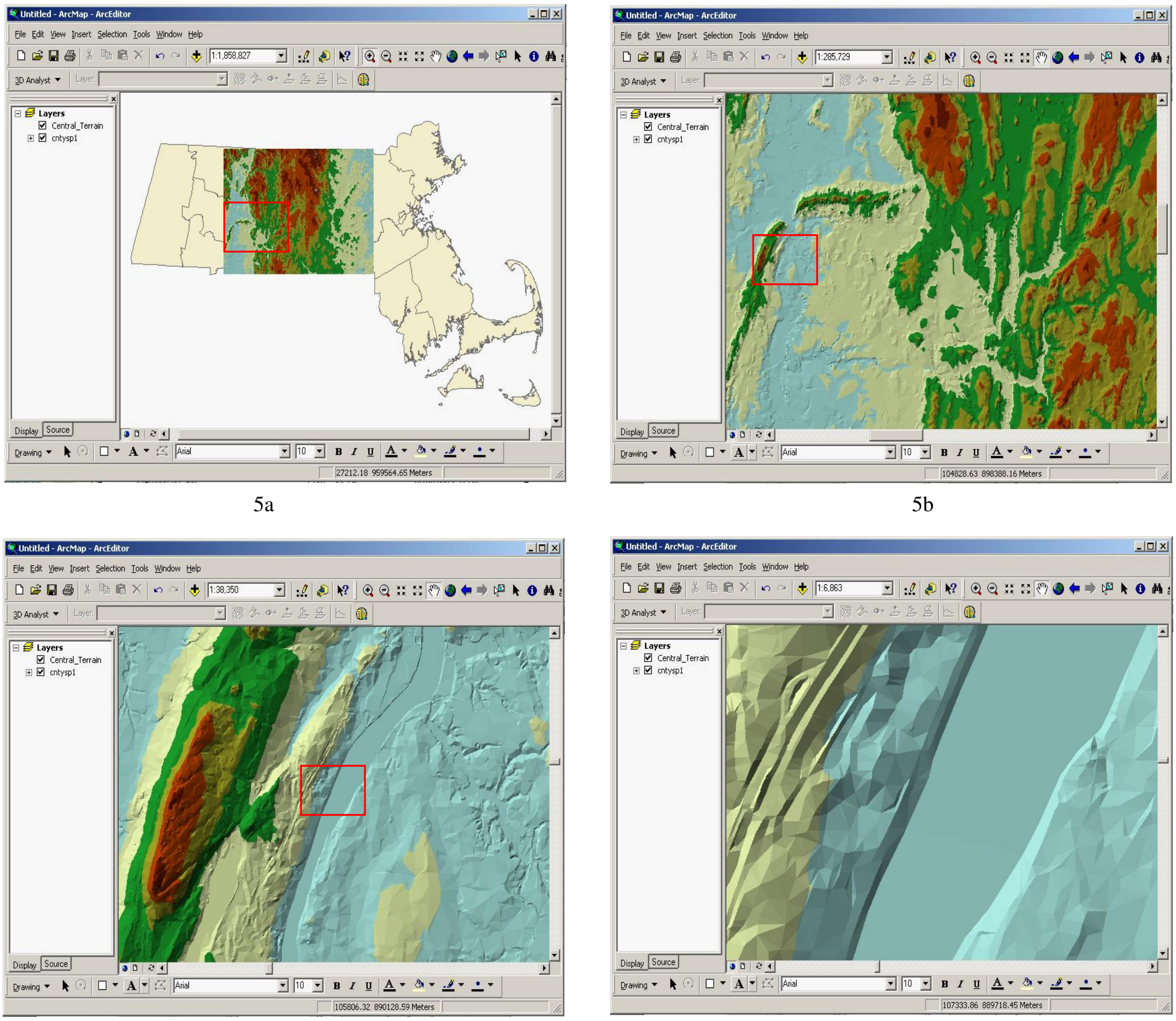


Figure 4: Examples of Terrain application.

An interesting example of this dynamic query is surface rendering. The zoom in and zoom out operations represent a typical scenario of multi-resolution queries (Figure 5). The shaded area in Figure 5a shows the center part of the state of Massachusetts in the US. The full resolution model corresponding to the area contains about 16 million points, covering an area of 8800 square kilometers (110km x 80km). Obviously, it is a waste to apply all the points when zoomed to full extent, as many of them may be mapped onto the same pixels of the screen. In this case, a well-calculated, simplified version of the DTM may suffice to provide a good overview of the terrain. This also reduces the time used in DTM generation and rendering.



5a

5b

5c

5d

Figure 5: An example of Terrain application in surface visualization (source data: courtesy of MassGIS, Commonwealth of Massachusetts Executive Office of Environmental Affairs).

The DTM shown in Figure 5a contains only about 60000 points and was generated in several seconds. When zooming into a sub-area of interest, more data will be needed in order to provide a more detailed view. However, in the meantime, the extent of the query area has become much smaller, resulting in higher resolution, but a potentially smaller (or an acceptably sized) DTM (31000 points in Figures 5b, 28000 points in Figure 5c, and 28000 points in Figure 5d). The *AOI* and ΔH can be calculated automatically for each zoom (and pan) operation. In this process, users can specify what level of vertical resolution to use at given scale by associating scale-ranges with pyramid layer indices.

Users can use the DTM generated as the result of a query to perform surface analyses. There may be, however, cases where a DTM cannot be created because of system constraints. This can happen if the query extent is too big, and high resolution is required, as in calculating volume and area, generating contours, profiles, and view-shed, all across the whole terrain extent. This problem can be solved by performing such tasks tile by tile (or a sub-group of tiles by a sub-group of tiles), and then unifying the results.

5. CONCLUSIONS

This paper has presented an efficient approach for GIS users to handle large terrain data and model surface applications. Because only measurements and rules are stored in a database, users can take the advantages of TIN and GRID structures without sacrificing storage or losing information. The tiling scheme makes it possible to perform large-scale tasks that require working on a DTM of high resolution. It also helps to achieve spatial coherence, thus speeding up spatial queries, and reducing disk I/O and network traffic. Vertical indexing provides another contribution to further speed up spatial queries.

Storing terrain data as feature classes in a feature dataset allows them to be integrated with 2D data and be shared by other applications, such as Topology and Geometric-Network (Zeiler, 1999). The ESRI geo-database framework also provides a foundation to implement versioning and multi-user access support. The Invalidated-Area mechanism provides an efficient and elegant way to detect changes to terrain measurements, and enables local updating.

The Massachusetts example presented in this paper, and other in house testing cases, have demonstrated the capability and efficiency of the proposed approach. The approach may also benefit the user who can now contract out for the source measurement data and handle it in an efficient and flexible manner for many applications, as opposed to contracting out a GRID model that is limited.

There are, however, several tasks that still require further research and development. These include 1) a better vertical indexing mechanism for line and area features, 2) on-the-fly 3D generalization of line and area features, and 3) "horizontal" multi-resolution queries involving line and area features.

The current design and implementation of the system has introduced some storage overhead in order to support tiling and vertical indexing. Efforts are also needed in order to minimize or eliminate the impact of this overhead.

REFERENCES

Floriani, D., "Multiresolution modeling in geographical information systems", Unpublished paper Presented at *GISRUK'95*, University of Newcastle: Newcastle-upon-Tyne, UK. (1995)

Gottshalk, H.J., "Die Generalisieerung von Isolinien als Ergebnis der Generalisierung von Flächen", Zeitschrift für Vermessungswesen, Vol. 97, No. 11, pp. 489-494. (1972)

Heller, M., "Triangulation Algorithms for Adaptive Terrain Modeling", proceedings of the *Fourth International Symposium on Spatial Data Handling*, Zurich, Vol. 1, pp. 163-174. (1990)

Hoppe, H., "Smooth View-Dependent Lecel-of-Detail Control and its Application to Terrain Rendering", *IEEE Visualization*, October 1998, pp. 35-42. (1998)

Johns, C.B. and Abraham, I.M., "Line Generalization in a Global Cartographic Database", *Cartographica* 24: pp. 32-45. (1987)

Kalvin, A.D., "Superfaces: Polygonal Mesh Simplification with Bounded Error", *IEEE Computer Graphics and Applications*, May 1996, pp. 64-77. (1996)

Kidner, D.B., Ware, J.M., Sparkes, A.J., Jones C.B., "Multiscale Terrain and Topographic Modelling with the Implicit TIN", in *Transactions in GIS* (eds J.P. Wilson, A.S. Fotheringham, P. Forer), Vol. 4, Issue 4, October 2000, pp. 379-408. (2000)

Lee, R.W., "Who Says Can't Teach an Old New Tricks", *IMAGE Conference*, Scottsdale, Arizona, August 2-7, 1998, 5p. (1998)

Loon, J.C., "Cartographic Generalization of Digital Terrain Models", *Doctoral dissertation*, The Ohio State University. Ann Arbor, Michigan: University Microfilms International, UMI 79-02171. (1978)

Peng, W., "Database Generalization: Concepts, Problems, and Operations", proceedings of the *XIX ISPRS Congress*, July 16-23, 2000, Amsterdam, The Netherlands, 8p. (2000)

Peng, W., "Automated Generalization in GIS", *PhD Thesis*, Wageningen Agricultural University and ITC, The Netherlands, 188p. (1997)

Peng, W., Pilouk, M., Tempfli, K., "Generalizing Relief Representation Using Digital Contours", proceedings of the *XVIII ISPRS Congress*, July 9-19, 1996, Vienna, Austria, 8p. (1996)

Reinhard, K., "Multiresolution Representations for Surfaces Meshes Based on the Vertex Decimation Method", Computer & Graphics, Vol. 22, No. 1, pp. 13-26. (1998)

Weibel, R., "Models and Experiments for Adaptive Computer-Assisted Terrain Generalization", *Cartography and Geographic Information Systems*, Vol. 19, No. 3, pp. 133-152. (1992)

Weibel, R., "Konzepte und Experiment eur Automatisierung der Reliefgeneralisierung", *Doctoral dissertation*, Geo-Processing Series, Vol. 15, Department of Geography, University of Zurich. (1989)

Wolf, G.W., "Generalisierung topographischer Karten mottels Oberflächengraphen", *Doctoral dissertation*, Department of Grography, University of Klagenfurt. (1988)

Wu, H., "Prinzip und Method der automatischen Generalisierung der Reliefformen", Nachrichten aus dem Karten-und Vermessungswesen, series I, Vol. 85, pp. 163-174. (1981)

Yoeli, P., "Entwurf einer Methodologie fur computergestütztes kartographisches Generalisieren topographischer Reliefs", Kartographisches Generalisieren, *Swiss Society of Cartography* (ed.), (Cartographic Publication Series, Vol. 10, Zurich. (1990)

Zeiler, M., "Modeling Our World – The ESRI Guide to Geodatabase Design", *ESRI Press*. (1999)

Zoraster, S., Davis, D., and Hugus, M., "Manual and Automated Line Generalization and Feature Displacement", ETL-Report, Vol. ETL-0359 (plus ETL-0359-1). Fort Belvoir, Virginia: U.S. Army Corps of Engineers, Engineer Topographic Laboratory. (1984)