ON THE INTEGRATION OF DIGITAL TERRAIN AND SURFACE MODELING INTO GEOGRAPHIC INFORMATION SYSTEMS

Robert Weibel
Department of Geography
University of Zurich
Winterthurerstrasse 190
CH-8057 Zurich (Switzerland)
weibel@gis.geogr.unizh.ch

ABSTRACT

Current GIS are still predominantly oriented towards the processing of planimetric (2-D) data. On the other hand, there is an increasing need for capabilities to handle 2.5-D and 3-D data in many GIS applications. Based on the example of digital terrain modeling, this paper discusses the requirements and possibilities for the functional and database integration of digital terrain models (DTMs) into GIS. Functional integration based on a unified user interface and file structures today is implemented in several GIS. Current solutions for database integration, however, are prohibitively inefficient and the useful properties of DBMS cannot yet be exploited given the current state of research in spatial databases. Future research should foremost focus on the development of more efficient database schemata for DTM data structures. Special consideration should be given to data structures that can be used for disk-based triangulation algorithms, as they can provide an equivalent representation in memory and secondary storage. A further problem is the transfer of DTMs between different systems. While the transfer of geometry and topological relations could be implemented relatively easily through an extension of current transfer standards, the transfer of special local adaptations and interpolation methods that model the surface is non-trivial. Methods of object-oriented databases that allow the definition of persistent objects could offer a solution.

INTRODUCTION

In recent years, an increasing need has been expressed in the literature for an extension of planimetric (2-D) geographic information systems (GIS) to accommodate 2.5-D and even 3-D modeling functions. Based on the example of digital terrain and surface modeling, this paper looks at the requirements and possibilities for integrating such functions into planimetric GIS at the functional and at the database level. Integration involves two aspects: (1) *functional integration*, the transparent use of 2-D and 2.5-D or 3-D functions under common data formats and a unified user interface; and (2) *database integration*, the permanent storage and management of DTM and planimetric data under a common data management scheme, possibly a common database.

A digital terrain model (DTM) provides a 2.5-dimensional digital representation of a portion of the Earth's surface. By 2.5-D it is commonly meant that the topographic surface is represented as a field, having unique z-values over x and y, rather than providing a true 3-D surface or volume representation. The concept of DTMs is not restricted to topographic surfaces, but can be used to represent other continuously varying phenomena such as stratigraphic bedding planes, air temperature, or population density. Because of their great importance in many GIS applications, DTMs are taken as an example in this paper to discuss the integration of non-planimetric functionality into planimetric GIS. However, many of the observations made here also apply to 3-D modeling, if at a higher level of complexity.

The problem of permanent storage of DTMs and their integration into GIS databases is rarely treated in the literature. Emphasis is still on memory-based algorithms and data structures rather than on the permanent storage of DTMs. However, given the growing size of current terrain modeling projects and the general trend towards the exploitation of database technology in GIS, problems of storage and management of terrain-related data gain increased importance.

The predominant approach for DTM integration is still based on file structures relating to the data structures used in memory, and coupling of terrain modeling functions with the other GIS functions through a common user interface. Thus, while DTMs are functionally rather well integrated into some GIS, database integration is most often lacking completely. More recently, some authors have advocated the full integration of DTMs into the database system of the GIS, which would potentially allow enhanced retrieval and querying functionality. While some of these papers (e.g., Fritsch and Pfannenstein 1992, Molenaar 1990, Höhle 1992) helped to cover some of the theoretical basis of DTM integration, they still fall short of providing practicable solutions. Various issues have to be considered in order to develop efficient schemes for data integration of DTMs into GIS that actually meet the functional requirements of different DTM applications.

Four main topics will be addressed here: Which data representations exist in a DTM system? What are the requirements of different applications with respect to DTM integration? What are the alternatives for DTM storage and integration? How can terrain-related data be transferred between different systems?

MODELING COMPONENTS OF A DTM SYSTEM

Functional Tasks of Digital Terrain Modelling

The following five general tasks can be distinguished within a digital terrain modeling system:

- DTM generation: Sampling of original terrain observations (data capture), and formation of relations among the diverse observations (model construction) to build a DTM.
- DTM manipulation: Modification and refinement of DTMs, and the derivation of intermediate models.
- DTM interpretation: Analysis of DTMs, information extraction from DTMs.
- DTM visualization: Display of elements and properties of DTMs and derived information.
- DTM application: Development of appropriate application models for specific domains (e.g., models for erosion potential, surface water runoff, or noise pollution) to make efficient and effective use of terrain modeling techniques.

These five tasks should be treated as interrelated components of the terrain modeling process; they form elements of a chain that can only be as strong as its weakest link. Thus, a comprehensive DTM system should attempt to support all tasks of terrain modelling equally well, and be adaptive to given problems.

The above scope of a DTM system defines the range within which functional requirements of individual application domains can be specified.

Data Representations within a DTM System

Usually, DTMs are associated with data structures such as regular grid, triangulated irregular networks (TINs), contours, or surface patches. However, besides these representations, other terrain-related models exist in a DTM system, each of them necessitating particular data structures and specific treatment.

The input for the generation of DTMs is given by original observations that are sampled as point, line, or polygon features (e.g., spot heights, breaklines, contours or dead areas). Since they form the foundation of the DTM, they can be termed the *basic model* (Weibel and Heller 1990). The management of these features in the GIS database is simple, as long as the database schema allows to handle z-coordinates for points, lines, and polygon outlines. The basic model reflects the best state of knowledge obtainable for a given project and is thus stored permanently. Any other representation in a terrain modeling project can be derived from this basis, if the rules for generating derived models are known.

Various analytical and visual products can be derived from DTMs through interpretation and visualization (e.g., drainage basins, viewsheds, contours, perspective views, etc.). The models that can be used to represent these products normally are equivalent to those used for corresponding 2-D data. Thus, functional and database integration of products derived from DTMs is easily possible. Again, however, it must be possible to represent z-coordinates in the database (e.g., to store the 3-D coordinates of drainage channels or basins).

DTMs turn the basic model into a description that is useful for surface modeling. Of the various data structures are possible for DTMs, the two classes of regular grid, and triangulated irregular network (TIN) data structures are predominant. Since most classes of data structures used for DTMs do not relate to those commonly employed for handling planimetric features, different solutions must be found for the storage and management of these models. The question then is how to integrate DTMs with other representations, specifically with the basic model. In other words, the problem is how to maintain consistency of representations across different models. Since different DTM applications have different requirements, it can be expected that alternative solutions are necessary for DTM integration.

DTM INTEGRATION: REQUIREMENTS

The general requirements of a DTM system that is used in conjunction with a GIS typically include a wide range of criteria (the terms used below are partly based on Meyer 1988):

- Functionality: the ability to support a wide range of applications. This is probably the single most important criterion for users interested in getting their job done.
- Efficiency: the good use of the available resources, both in terms of storage space and computing time. This criterion is especially important when large amounts of data need to be processed or fast response times are required for high interactivity. DTMs today easily involve several 100,000 data elements.
- Correctness: the ability to deliver a correct solution for a given problem (or, alternatively, inform the user about the inability to do so).
- Consistency: the ability to avoid or resolve conflicts between multiple representations of an entity. A conflict may, for instance, arise when an element in the basic model is removed, but this change is not propagated to the DTM which had been generated previously from these input data.
- Robustness: the ability to resolve special cases in the input data or at least fail gracefully. This requires sound exception handling strategies.
- Compatibility: the ability to transfer data between different representations or systems, minimizing information loss. This issue mainly relates to the transfer of complex data structures (e.g., TINs) between different systems.
- Adaptivity: the ability to adapt to surface character and modeling purpose. This requirement also necessitates the ability to support different representations of a phenomenon – different data structures or interpolation schemes – and to transform between different representations.
- Extendibility: the ease with which the data may be adapted to changes in specifications. Updates of the data may occur over time, necessitating version

management strategies and procedures to integrate different states into a single coherent and consistent model.

- Security: the ability to control access to the data selectively for individual users. This aspect is not essential for all applications. It may be sufficient to rely on the security mechanisms of the operating system.
- Concurrency: the ability to manage data access of multiple users at the same time. This criterion can be important for large installations with many concurrent users.
- Ease of use: the ability to let users interact with the software in an intuitive and flexible way. Significantly contributes to the performance users can get from the system in terms of functionality, efficiency, and correctness.
- Ease of integration: the ability to integrate DTMs as well as products derived thereof with other data in a GIS. This criterion has a major impact on the functionality and ease of use of a GIS package, and also influences the efficiency of workflows involving multiple steps.

Of course, the actual requirements which are expected from a specific system depend on the particular target applications and are usually only a subset of the above list. Some criteria are essential for all applications: ease of use and efficiency are always desirable, correctness and robustness are always required, and adaptivity to varying data characteristics is fundamental to ensure correctness. Consistency is important in the context of DTMs, because they always involve multiple representations (basic model, grids, TINs, etc.).

Some applications have very specific requirements. For instance, a DTM system for use in opencast mining requires efficiency in order to keep track of changes of the terrain surface, and necessitates extendibility and consistency of multiple representations of the surface at different points in time. An institution which is involved in data production and distribution assigns high priority to the compatibility of DTM data formats. Security and concurrency are commonly restricted to environments where controlled multi-user access is required (e.g., large institutions with personnel from different departments working concurrently on the same data).

With respect to the criteria relating to the use of DBMS for managing DTMs – consistency, extendibility, security, and concurrency – two broad classes of application domains can be distinguished. The first class consists of applications that focus on DTM generation and manipulation, tasks that are usually important to institutions collecting, editing, managing, and distributing terrain-related data. The second group of applications concentrates on information extraction, that is, on DTM interpretation and visualization. Examples of such uses are the computation of gradient and aspect information (e.g., for the estimation of the potential for soil erosion) or the production of contour maps or block diagrams. These applications typically operate as batch-like processes, and do not require access to individual elements of a DTM.

While the first class of applications could directly benefit from a database integration of DTMs into GIS, requirements of the second class with respect to DTM storage and management are rather limited. For instance, it is very rare that a user would want to know the gradient of a particular triangle of a TIN.

DTM INTEGRATION: POSSIBLE STRATEGIES

Maintenance of Consistency

As noted above, the maintenance of the consistency between the basic model and derived DTMs is a major concern, independently of the data structures (grid, TIN) that are used to represent the DTMs. Inconsistencies may be introduced whenever modifications to the shape of the terrain surface become necessary.

Possible cases include the addition of new terrain observations to improve the surface model, or the removal of erroneous data elements.

If both the basic model and the derived DTMs remain static and no changes occur, there is no danger that the consistency between the different representations is violated, even if they are stored independently from each other. However, if modifications to the contents of one or the other of the models are made, the surface will be represented differently depending on which model is used, and steps have to be undertaken to prevent or eliminate these discrepancies. Usually, the modification of a single feature in the basic model will induce changes to more than one element of the associated DTMs, since surface gradients in the vicinity of the affected feature will also change.

A practical approach to maintaining consistency is to regard DTMs (grids, TINs) as derivative products from the basic model, and to focus all editing operations on the elements of the basic model. The DTMs are used as a visual backdrop in the editing process (e.g., via contours, hillshading, or stereo) to give an impression of the form of the continuous surface, which the basic model obviously cannot. This approach greatly simplifies the maintenance of consistency between models. Change propagation occurs from one model to many, instead of from many to many models. Changes can then be propagated from the modified basic model to the associated DTMs through either global regeneration (building the entire models anew) or, preferably, local modifications involving only those elements which are affected. Functions for editing of DTMs are discussed in Weibel and Heller (1991).

Alternatives for DTM Integration

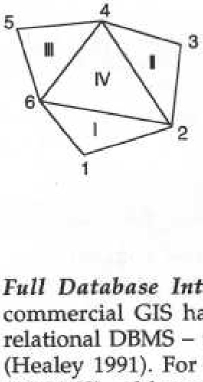
Loose coupling: The simplest form of DTM integration consists of a loose coupling mechanism between the DTM system and the GIS via common data interfaces. While the two systems operate independently, this mechanism allows to export data sampled in a GIS to the DTM system to generate a surface model, and import back the results of surface analyses. This approach is frequently found in GIS that do not dispose of a built-in terrain modeling module, but offer an interface to DTM software systems provided by other vendors instead. The most serious drawbacks of this strategy are that queries across systems are difficult and inefficient, communication is usually restricted to the exchange of entire data sets, and inconsistencies can easily be introduced between representations in the two systems.

Functional Integration: The next step is to incorporate the software module for terrain modeling into the GIS under a unified user interface and common data formats, offering better opportunities for functional integration.

This approach first of all has implications with respect to the user interface. While most batch-like functions can be handled by the same user interface designs as for 2-D GIS, interactive operations require specific control mechanisms. Examples include special manipulation controls for the interactive manipulation of projection parameters (viewpoint, view direction, camera parameters, etc.) of perspective displays and flight path design for animations (Hussey et al. 1986), or interaction tools for interactive surface and volume modeling and sculpting (Galyean and Hughes 1991). Depending on the complexity of the modeling operations that are involved, these 3-D control mechanisms are not just simple extensions of their 2-D equivalents.

In terms of storage schemes that are used to functionally integrate DTM systems into GIS, file-based structures are still the predominant method used today. The simplest file-based approach stores the entire information of a DTM in a single file. Usually, however, this approach is only practical for simple gridded DTMs. More complex representations such as TINs are predominantly handled in multifile structures, with multiple files relating to internal data structures such as the winged triangle structure (Fig. 1) which is used by various commercial systems. Usually, some header information is added to the files or an additional header file is created, including descriptive information which summarizes the characteristics of the DTM such as the number of vertices, number of triangles, number of hull vertices, number of breaklines, spatial extent, and data source. Figure 1 shows an example of a schema comprising three files that would typically result using the winged triangle data structure. Optionally, a header file with descriptive information and a file containing the vertices of the triangulation hull can be included. If programming languages are used that allow positioning in the byte stream of a file (e.g., Fortran, C, or C++), then the storage, retrieval and update of individual vertices or triangles is possible. While the winged triangle representation is widely used in TIN-based systems, it should be noted that other data structures are possible which are more compact or more efficient, or lend themselves better to disk-based algorithms (Heller 1990).

In case the carrier GIS makes use of a DBMS to store spatially referenced data, file-based storage schemes can easily be extended by storing descriptive information about DTMs in the database. That is, the actual DTMs are stored in single or multiple files, while the files names (including the path to their location in the file system) and the descriptive statistics are held in the database. This descriptive information can easily be stored as attributes in a DBMS. Via the query language of the GIS, DTMs can be retrieved, and queries can be answered such as 'Which DTMs of resolution better than 30 m are available within the city limits of Zurich?'. However, access through the DBMS is still restricted to DTMs as a whole. If individual elements of terrain models are to be retrieved, this must be accomplished by specific functions embedded in the DTM module. Also, advantageous features of DBMS such as security and recovery mechanisms cannot be exploited.



Triangle adjacencies

tri_nr	neighbors
I	0, IV, 0
II	0, 0, IV
III	0, 0, IV
IV	II, III, I

Vertices

tri_nr	vert_nr
I	1, 2, 6
II	2, 3, 4
III	4, 5, 6
IV	2, 4, 6

Vertex coordinates

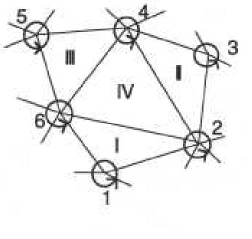
vert_nr	x	y	z
1	x1	y1	z1
2	x2	y2	z2
3	x3	y3	z3
4	x4	y4	z4
5	x5	y5	z5
6	x6	y6	z6

Fig. 1: Winged triangle data structure

Full Database Integration: Over the past decade, the majority of developers of commercial GIS have turned towards the use of commercial DBMS – primarily relational DBMS – to support the storage and retrieval of spatially referenced data (Healey 1991). For simplicity, the discussion here is restricted to relational DBMS (RDBMS), although other types exist (Healey 1991). Modern RDBMS offer a number of useful capabilities such as built-in query languages for retrieval of data elements, facilities for report generation, security mechanisms to limit access for unauthorized users, multi-user support, and rollback recovery mechanisms in case of system failure. On the other hand, the use of database technology for spatial data requires several important extensions to be made to standard RDBMS, relating to query languages, access methods, and database schemata (Frank 1988, Healey 1991, Haas and Cody 1991).

With respect to the integration of DTMs into RDBMS, the single most critical issue is probably the design of the database schema, since the time needed to access individual DTM elements increases with each table that is added and with each relational join that needs to be formed between tables. As a prerequisite to DTM integration, the database schema must be capable of holding x, y, *and* z-coordinates for the point, line, and polygon features of the basic model. (At this point, it should be noted that several commercial GIS products still do not meet this requirement, in which case 'workarounds' must be provided by storing spot heights, breaklines, and dead areas in text files.) The further discussion is restricted to database schemata for TINs, which are more complex to handle than grids. Possible solutions for grids are presented in Waugh and Healey (1987).

To store a TIN in a GIS database, it would be possible to treat triangles as a special case of polygons, with straight arcs and nodes that have z-coordinates (Waugh and Healey 1987). However, this strategy could only serve as a brute-force approach, as the specific topological properties of triangulations are not adequately modelled. Another possibility is to base the schema on the winged triangle data structure of Figure 1. Instead of the three files that were used for the file-based storage method discussed above, three tables are defined for the triangles, vertices, and vertex coordinates. This schema is, for instance, used in the system described by Steidler et al. (1990). A benefit of this scheme is that information about the individual TIN elements (topological relations and coordinates) can be readily queried. A further advantage is that points of the basic model and vertices of the TIN are made equivalent: they are stored only once in the table representing the codes of the sample points and TIN vertices, respectively. This mechanism guarantees that if a vertex is removed from the TIN, the corresponding point is also removed from the basic model. The reverse, however, is not true. If a point is added in the basic model, the TIN is not automatically updated. This schema can also not enforce the consistent representations of lines or polygon outlines (e.g., breaklines) in both the basic model and the TIN. All of these operations have to be taken care of by the application software (i.e., by a specific DTM editing module), since the database mechanisms cannot provide the appropriate consistency checks. Another major drawback of this schema is its inefficiency. Since the information pertaining to a single triangle is distributed over three tables, numerous relational joins are necessary when this information is accessed at run time. While the winged triangle structure is not inefficient as an internal data structure, it is slow when used for database storage, slowing down performance by orders of magnitude in comparison to file-based storage.



Vertices				
vert_nr	x	y	z	neighbors
1	x1	y1	z1	2, 6, 17, 19
2	x2	y2	z2	1, 19, 22, 25, 3, 4, 6
3	x3	y3	z3	2, 25, 31, 4
4	x4	y4	z4	2, 3, 31, 39, 5, 6
5	x5	y5	z5	4, 39, 42, 43, 6
6	x6	y6	z6	1, 2, 4, 5, 43, 17

Fig. 2: Vertex-oriented data structure

A more compact schema is shown in Figure 2. It is based on a vertex-oriented data structure, needing only one table to fully represent the TIN topology. For each vertex, its adjacent vertices are stored in a counterclockwise order. Thus, a given vertex plus any two consecutive neighbors define a triangle (except for vertices on the hull of the triangulation). Since the number of neighbors for each vertex may vary (but is usually around 5 to 6), the DBMS must provide fields of variable length (bulk data type); this requirement is not met by all commercial RDBMS. This vertex-oriented schema performs much better in terms of database storage and retrieval than the winged triangle approach. On the other hand, it is also not very useful as an internal data structure for surface modeling, and is usually transformed into triangle or edge-oriented representations (Heller 1990). While transformations between different TIN data structures are possible, they slow down the overall performance of the terrain modeling operations.

Many practical DTMs are based on models that are much too large to fit into the memory of any computer. Even if a generously equipped machine could handle the entire model internally, it would be quite inefficient to always read in the entire data set if only a small portion of the model is needed. This is a very realistic scenario as it is most likely that complex models are not built in one run, but in progressive, incremental steps. At each refinement step only a locally confined part is accessed and modified. It is therefore quite important to be able to efficiently access local subsets on the external storage. It must be attempted to structure the model spatially and organise the data on disk according to their spatial configuration. Furthermore, algorithms must be designed so they can profit from the spatial structuring. The data that is typically needed concurrently should be stored correspondingly on disk. Thus, it is possible to substantially reduce the disk-access for frequent operations. Of course, further improvements can be achieved by not storing data that can be reconstructed more efficiently than read. As the research of Heller (1993) has shown, using a hierarchical triangulation scheme on dynamically split Delaunay buckets with a compact, point-based but edge-oriented data structure leads to a very promising solution for disk-based triangulation methods.

TRANSFER OF TERRAIN-RELATED DATA

The portability of DTMs and associated data is a very practical issue. The ease with which the transfer of terrain-related data between different systems can be achieved mainly depends on the complexity of the relations which are modelled by the DTM and on the functionality of the sending and the receiving DTM system. While the transfer of digital cartographic data has received wide attention in the past few years, culminating in the publication of national standards such as the Spatial Data Transfer Standard (SDTS) in the USA (DCDSTF 1988), the transfer of DTMs between different systems can still pose considerable problems.

Because of their simple structure, data transfer can be accomplished relatively easily for gridded DTMs. Grids can be treated like raster data, including additional information about grid spacing, the spatial extent of the DTM, the source of the data and other descriptive information. One possible exchange format that is used by the US Geological Survey for the distribution of their DTM product called 'Digital Elevation Model' (DEM) is described in USGS (1987). SDTS offers a more generic exchange mechanism for grids and rasters.

For the feature data making up the original observations of the basic model, the same data transfer formats can be used as for point, line, and area features basically. However, it is crucial in this case that the third dimension – that is, the z-coordinates of the points and lines – can be transferred as well. While most data transfer formats are focusing on purely planimetric cartographic data, more generic formats such as SDTS can accommodate the transfer of z-coordinates.

For TINs, no published transfer format exists to our knowledge. Since TINs are a more generic structure than grids, various alternative data structures exist, and most DTM software packages use different internal data structures as well as storage structures. In order to transfer the complete information contained in a TIN, the exchange mechanism not only must be capable of transferring the TIN geometry, topology, and descriptive information (e.g., information about data quality). Additionally, a TIN may also be locally adjusted to account for breaklines or contour data, and modified gradients may be inferred to honor surface discontinuities. Finally, the interpolation method that is used is of great importance for determining the actual shape of the modelled surface.

While it is possible to exchange the geometry and descriptive information of a TIN using comprehensive transfer standards such as SDTS, transferring the specific topological relations of a triangulation can only be accomplished in a rather inefficient way (e.g., by treating triangles as polygons). The only available mechanism of 'transferring' TINs currently is therefore to exchange the original observations between systems and rebuild the TIN topology in the receiving system. This may be feasible if the same form of triangulation (e.g., Delaunay) can be used in both systems. However, if the triangulation has been modified in any way or if the interpolation methods are not equal in both systems it is unlikely that an equivalent TIN can be reconstructed in a different system. Thus, the comparability of results of operations (e.g., visibility analysis or gradient calculation) generated from different DTM systems is hardly possible.

It would be relatively simple to extend standards like SDTS by definitions for TIN topology such as the ones shown in the previous section. It would thus be possible to transfer the TIN geometry and topology. However, as mentioned above, a complete definition of the surface also includes rules for specific local adaptations of the TIN and interpolation methods. A possible solution for the transfer of these procedural elements could be the standardization of construction and interpolation methods. However, this approach would be unrealistic given the wide range of available methods and the diversity of potential uses. A more promising approach is to make use of techniques for the persistent storage of methods as they are available in object-oriented databases. One example is the commercial object-oriented DBMS ObjectStore (Object Design 1991), which allows to implement persistent objects, complete with methods, in the object-oriented programming language C++. The use of common, standardized notations such as C++ may thus provide a mechanism for the exchange not only of data, but also of methods between different GIS.

CONCLUSION

This paper has sought to discuss the requirements and different strategies with respect to functional and database integration of DTMs into GIS. The discussion has shown that while theoretically, integration of DTMs into the DBMS of GIS would have benefits such as increased data security and multi-user access control, these useful properties cannot yet be exploited given the current state of research in spatial databases. Most of the advantages that DBMS can provide for attribute data, such as consistency checking and query operations, must be handled by the application software in the case of spatial data, especially for DTMs. Because database schemata currently used to represent DTMs are inefficient, DBMS currently are not much more than more secure, but also a lot less efficient storage systems in comparison to file-based storage. Presently, DTMs integrated into GIS software systems thus take a practical approach focusing on functional integration: the basic model is stored in the GIS database, and the DTMs are derived thereof and are stored in file structures. As an extension, descriptive information about the DTMs could be stored in the DBMS, allowing queries relating to entire models. As long as tasks are involved that require only little interactivity, modifications to the user interface and query language are only minor.

Nevertheless, the general trend is leading towards better database integration, for planimetric as well as for surface data due to requirements for increased data volumes and interactivity. Future research should foremost focus on the development of more efficient database schemata for DTM data structures. Special consideration should be given to data structures that can be used for disk-based triangulation algorithms (Heller 1993), as they can provide an equivalent representation in memory and secondary storage. Finally, the transfer of DTMs and related data remains a problem, requiring increased attention in future attempts to develop transfer standards that extend beyond cartographic (i.e., planimetric) data.

ACKNOWLEDGEMENTS

I would like to thank my colleague Martin Heller for fruitful discussions and for reviewing the manuscript.

REFERENCES

DCDSTF (Digital Cartographic Data Standards Task Force) 1988, The Proposed Standard for Digital Cartographic Data. *American Cartographer,* **15**(1): 9-140.

Frank, A.U. 1988, Requirements for a Database Management System for a GIS. *Photogrammetric Engineering and Remote Sensing,* **54**(11): 1557-1564.

Fritsch, D. and Pfannenstein, A. 1992, Integration of DTM Data Structures into GIS Data Models. *International Archives of Photogrammetry and Remote Sensing,* **29**(B3): 497-503.

Galyean, T.A. and Hughes, J.F. (1991). Sculpting: An Interactive Volumetric Modeling Technique. *Computer Graphics.* (SIGGRAPH '91), **25**(4): 267-274.

Haas, L.M. and Cody, W.F. 1991, Exploiting Extensible DBMS in Integrated Geographic Information Systems. In: Günther, O., and Schek, H.-J. (eds.), *Advances in Spatial Databases,* Lecture Notes in Computer Science **525**. Berlin: Springer-Verlag, 423-450.

Healey, R.G. 1991, Database Management Systems. In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.). *Geographical Information Systems: Principles and Applications,* London: Longman, **1**: 251-267.

Heller, M. 1990, Triangulation Algorithms for Adaptive Terrain Modeling. Proceedings Fourth International Symposium on Spatial Data Handling. Zurich, Switzerland, **1**: 163-174.

Heller, M. 1993, A Synthesis of Triangulation Algorithms. Technical Report. Department of Geography, University of Zurich, August 1993, 22 pgs.

Höhle, J. 1992, The Object-Oriented Height Model and Its Application. *Int. Archives of Photogrammetry and Remote Sensing,* **29**(B4): 869-873.

Hussey, K.J., Hall, J.R. and Mortensen, R.A. 1986, Image Processing Methods in Two and Three Dimensions Used to Animate Remotely Sensed Data. *Proc. IGARSS '86 Symposium.* Zurich, 8-11 September 1986, **2**: 771-776.

Meyer, B. (1988). Object-Oriented Software Construction. New York: Prentice-Hall.

Molenaar, M. 1990, A Formal Data Structure for Three Dimensional Vector Maps. *Proceedings EGIS '90.* Amsterdam, The Netherlands, **2**: 770-781.

Object Design, Inc. 1991, *ObjectStore User Guide.* Burlington.

Steidler, F., Dupont, C., Funcke, G., Vuattoux, C., and Wyatt, A. 1990, Digital Terrain Models and Their Incorporation in a Database Management System. In: Petrie, G., and Kennie, T.J.M. (eds.), *Terrain Modelling in Surveying and Civil Engineering.* Caithness, UK: Whittles Publishing Services, 209-216.

USGS 1987, Digital Elevation Models. *US Geological Survey Data Users Guide 5.* Reston, VA: USGS.

Waugh, T.C. and Healey, R.G. 1987, The GEOVIEW Design: A Relational Data Base Approach to Geographical Data Handling. *International Journal of Geographical Information Systems,* **1**(2): 101-118.

Weibel, R. and Heller, M. 1990, A Framework for Digital Terrain Modeling. *Proceedings Fourth International Symposium on Spatial Data Handling.* Zurich, Switzerland, **1**: 219-229.

Weibel, R. and Heller, M. 1991, Digital Terrain Modelling. In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W. (eds.). *Geographical Information Systems: Principles and Applications,* London: Longman, **1**: 269-297.