Towards a hybrid framework for the visualization and analysis of 3D spatial data

Alejandro Graciano
Department of Computer Science, University of Jaén
Jaén, Spain
graciano@ujaen.es

Antonio J. Rueda
Department of Computer Science, University of Jaén
Jaén, Spain
ajrueda@ujaen.es

Lidia Ortega
Department of Computer Science, University of Jaén
Jaén, Spain
lidia@ujaen.es

Francisco R. Feito
Department of Computer Science, University of Jaén
Jaén, Spain
ffeito@ujan.es

ABSTRACT

Several geospatial problems like urban subsurface analysis involve data from multiple domains and nature. For instance, an urban infrastructure analysis must take into account not only the urban elements but their geological environment. This requires the definition of hybrid schemes and algorithms that keep the dimensionality and domain of the data, while allowing the joint management of both representations in a combined way. In addition, a proper 3D visualization method that shows those heterogeneous data together can be very useful for geoscientific and GIS professionals. In this paper, we present the foundations of a real-time 3D visualization framework capable of rendering field and vector data, as well as a set of operations that can solve many problems for engaging data from different domains. We propose the use of the Stack-Based Representation of Terrains for field data which provides a whole 3D representation of volumetric terrains while allowing an efficient memory usage. The resulting hybrid framework can help geoscientists and engineers to analyze 3D complex geospatial data and make decisions at a glance.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**; • **Computing methodologies** → *Scientific visualization*; *Volumetric models*; • **Applied computing** → *Environmental sciences*;

KEYWORDS

3D GIS, Geomodelling, Geovisualization, Hybrid GIS, Stack-Based Representation

ACM Reference Format:
Alejandro Graciano, Antonio J. Rueda, Lidia Ortega, and Francisco R. Feito. 2017. Towards a hybrid framework for the visualization and analysis of 3D spatial data. In *UrbanGIS'17: UrbanGIS'17:3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics , November 7–10, 2017, Redondo Beach, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3152178.3152183

UrbanGIS'17, November 7–10, 2017, Redondo Beach, CA, USA

ACM ISBN 978-1-4503-5495-0/17/11…$15.00
https://doi.org/10.1145/3152178.3152183

1 INTRODUCTION

Geographic Information systems (GIS) are key elements in the development of Smart Cities. GIS give valuable tools for collecting data, decision making, data visualization and other stages of the urban planning and management. Traditionally, GIS have been largely rooted in 2D space. This dimensionality can be enough for many applications. However, advances in data acquisition [18] have provided geoscientists with huge amounts of heterogeneous multidimensional data that can only be visualized and analyzed with new methods and tools in 3D or 4D space.

Recently, the research and development of 3D GIS have attracted the interest in many different fields [16] [50]. However, the management of the 3D spatial information, and particularly its 3D visualization, is not yet mature enough. 3D GIS are normally limited to representing terrains as 2.5 Digital Elevation Models (DEMs), without any or little volumetric information, which has great importance in many geoscientific fields [21]. Even though some 3D GIS packages contain voxel models as basic data type, its usefulness is limited to a 3D extension of common raster algorithms [14] [33].

Frequently, spatial problems may involve data with different nature [5]. This is another shortcoming of current GIS tools, since although most GIS can deal with vector and field representations, they lack of hybrid analysis methods that operate with both representations at the same time. For instance, an engineering survey may require resistivity terrain data modeled as a field, as well as geometric data, like a point cloud provided by a LiDAR scanner or information about surrounding pipelines, in order to estimate the cost of installing new structures. In these surveys we find the use of systems based on CAD/CAM [7] or GIS extensions, in most cases quite expensive, that only solve very specific problems [11]. In the literature there also exist works that perform conversions between different data types in order to work with a single data representation [41].

When converting data such as material terrain distribution from field data type to vector data type, it is necessary to extract geometric entities. Most approaches use B-rep schemes to perform this conversion [20] [47]. This type of representation may generate models with redundant geometry like the horizons between strata

layers. In addition, internal structures (e.g., aquifers, petroleum reservoirs among others) are more difficult to represent and visualize than those based only on volumetric data. Furthermore, a conversion from vector to raster data implies a loss of resolution in the discretization process [1].

Another different approach for dealing with field and 3D vector data in an integrated way is by means of the definition of a general or hybrid representation scheme that covers both data types. This topic has been under active research in the past 30 years [6] [48] [12]. However, as far as we know, the proposed formal schemes have not been fully implemented in current GIS or geoscientific applications.

In this paper we present a 3D framework for an efficient visualization and analysis of volumetric terrain models at surface and subsurface levels as well as vector spatial data. The nature of this information is always maintained. In addition we define several spatial operations that require inputs of both types such as volume analysis. The terrain is defined by a Stack-Based Representation (SBRT) [2]. This representation is an efficient extension of DEMs that allows more than a single elevation value per raster cell. Each cell of the SBRT has a vertical sequence of intervals, each of them containing a property (i.e., a terrain material) and a height. It can be seen as a compact representation of voxel models in which each interval encodes a series of voxels with the same xy coordinate and property value. The model is also capable of incorporating vector layers consisting of 3D geometric entities like pipeline networks or underground pits. Figure 1 shows an overview of our framework. It should be noted that the central element of the system is the combined visualization.

The remainder of the paper is organized as follows. Section 2 provides a review of the existing literature. Aspects about the representation models used in our framework are outlined in Section 3. Section 4 describes the visualization procedure implemented in our system as well as a performance analysis is discussed. Section 5 is focused on the explanation of the operations that our framework provides. Finally, we conclude our work in Section 6.

2 RELATED WORK

The modeling of 3D discrete or vector objects is an issue extensively studied, being one of the primary topics faced in GIS research. This topic has been approached both from the applied and formal points of view. Regarding the former, many researchers have used GIS tools to solve concrete case studies. For example, Zhang et al. [49] used ArcGIS in order to develop an augmented reality system for the management of underground pipeline data. In another work presented by Pueyo and Patow [35], 3D buildings are generated from 2D cadastral information. In [38], a 3D modeling software is used to represent 3D objects instead of 3D GIS due to their lack of flexibility. From a formal perspective, the approaches presented by Molenaar [27], Pilouk [32] or Chen [6] introduced data structures for representing discrete entities. These formal frameworks have inspired many other works [31] [8].

The large variety of models defined for the representation of discrete spatial data in urban contexts, involves the necessity of a general standard. For this purpose, the Open Geospatial Consortium defined CityGML [15]. Examples of the use of this standard include the paper introduced by Van den Brink et al. [46], in which they propose an approach for modeling CityGML documents from UML. Another example is the construction of 3D city models without information about elevation suggested by Biljecki et al. [3]. One of the main drawbacks of CityGML is that it is not suitable for the representation of volume-based models since it does only define objects boundaries, therefore, many geological features cannot be modeled among others. In [42], the authors describe an extension for CityGML to solve this issue. However, there are other papers focused on the representation of 3D cities in combination with 2D data with a custom representation of the spatial data, being their scope the modeling of the cities above the surface [10] [30].

In 2D and 2.5D GIS, field data represent mainly the surface of the terrain defined as a raster. Kreuseler [23] suggested an approach for visual analysis of 'real' 3D landscape representations, however, as the method is based on height maps, models with overhangs or caves cannot be represented. Nevertheless, there are some results focused on the representation of volumetric field data, mainly aimed at geological rather than urban environments [28] [26].

Furthermore, many papers about the combination of vector features and terrain surface have been presented. In most of them, a single overlapping procedure between both layers is performed. In the visualization tool presented in [44] many groundwater models are integrated to the terrain surface. There are other papers that overlay 2D cadastral data and the terrain layer, such as in [22], [5], [43]. More recently we find similar results in [45] and [40].

Commercial and open source GIS do not provide hybrid analysis operations that deal with raster and vector data at the same types, keeping their properties and nature such as QGIS [36] or ArcGIS [9]. For example, in QGIS it can be just performed simple statistics and clipping operations involving both data types, however more complex analysis operations are not available. In order to compute analysis operations, the input layers must be homogeneous, therefore a conversion between the different data types has to be computed. In addition, there were many attempts to develop a complete 3D visualization, especially in this hybrid scenario, improving the classical 2.5D visualization. Unlike current GIS, we suggest a system that maintains both a 3D visualization and analysis for raster and vector data types in a totally integrated framework.

3 DATA REPRESENTATION

Traditionally, the process of describing a new GIS or geospatial framework involves a four-level scheme [25] [24], whose levels describe a different abstraction of the world. A reality level (1) defines the objects of the real world that need be studied or managed. The real objects modeled in this particular case are divided into two categories: environmental information, mainly characterized by geological material distribution of the terrain, as well as urban infrastructures such as sewage systems. The conceptual level (2) is a type of abstraction that provides mathematical models for the representation of these real world objects in a computational discrete model. Traditionally, GIS systems define in this level the vector (urban entities) and field (material distribution) data types. The abstraction provided by common GIS layers belongs to this level. The logical level (3) assigns to the conceptual models concrete

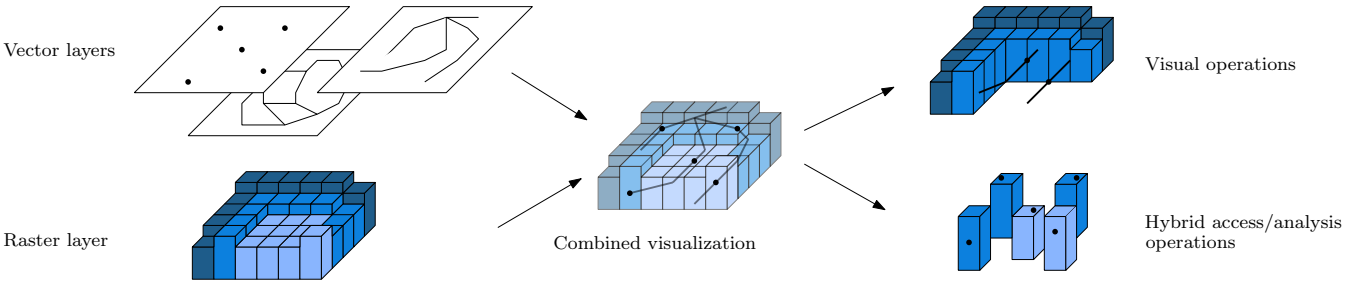

Figure 1: Overview of the framework suggested in the paper

representations or data structures. Finally, in the physical level (4), it is given a translation of the logical representations to a storage model, either in a file or database management system.

The following, is a more detailed discussion about the representations and data structures used in the conceptual level in our framework.

3.1 Logical model

This level provides details about data structures and representations such as the B-Rep and the SBRT schemes that were defined in an intuitive way in Section 1.

Formally, a SBRT can be defined as a 2D regular grid partition of space in which each cell, called stack St_i, is formed by a series of intervals $< m_j, h_k >$, where m_j corresponds with the material sampled in the interval, and h_k with the maximum height of the interval. Within the materials set we can include other values which are not considered strictly geological materials like air. This allows the modeling of common volumetric features present on the surface, such as overhangs, arcs or caves.

In our framework, all the discrete objects like points, segments, polygons, solids and segment networks are represented by B-rep models. Each discrete layer is considered basically a combination of 3D points p. For example, each element of a *segment layer* is formed by a tuple $s =< p_1, p_2 >$, a *polygonal layer* is characterized by $a =< p_1, p_2, \ldots, p_n >$., a *solid layer* is defined as a set of polygons $sl =< a_1, a_2, \ldots, a_n >$ and finally, a *segment network* is defined as series of segments and subnetworks in a recursive manner $nt =< s_1, s_2, \ldots, s_n, nt_1, nt_2, \ldots, nt_n >$. Figure 2 shows an example of the vector data types of our framework.

Each layer of our system contains a single data type and it keeps its nature at any time, i.e., a segment layer cannot include other data types than points, segments, networks, polygons or solids. For instance, a layer that represents the Internet network system of a building, can contains the wiring (segment network), nodes representing routers or switches (points) or polygons representing the signal zone of a wireless node.

In order to accelerate spatial queries and operations, the vector layers are organized using an octree as spatial index. Access to raster data in the SBRT is also very efficient as the division of the XY plane into regular cells with a fixed size defines a 2D spatial hashing for sampling stacks. Once the stack is retrieved, it must be traversed in order to find the value that contains a z coordinate. This traversal can be considered $O(1)$ in comparison with the total

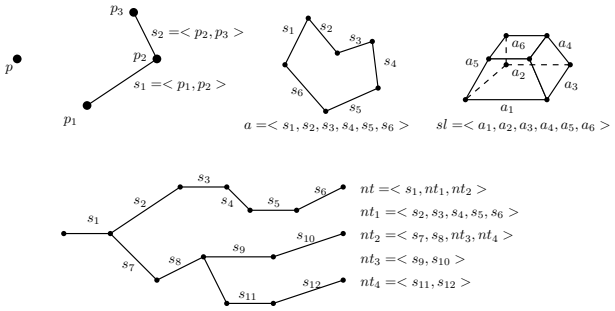

Figure 2: Vector data entities

amount of data handled, since the stacks in the subsurface terrain models usually never exceed 20 intervals.

4 VISUALIZATION

Any application that deals with spatial data in general and more specifically hybrid and heterogeneous data, requires an efficient, flexible and highly descriptive visualization that assists in the decision-making procedures.

The visualization of each layer uses different approaches depending on the layer type. On the one hand, vector layers are rendered using standard computer graphics methods. Each entity is assigned a triangle mesh model which is rasterized by the rendering engine. For instance, for each segment in a network layer a cylinder is chosen as visual representation, while in a point-based layer, well or manhole 3D models are used.

The approach used for the visualization of field layers is based on the well-known GPU raycasting algorithm for direct volume rendering. The key idea of the algorithm is to cast rays from the virtual camera through every pixel in the framebuffer, combining the color values sampled from the data along the ray [19]. Commonly, volumetric datasets are arranged in 3D textures to send them to the GPU, but in our case we use a different memory layout. As depicted above, the procedure of sampling a SBRT is divided into two steps: the stack identification and its iteration. Therefore, the memory scheme we propose is based on two structures: a spatial index structure and an interval structure where the stacks descriptions are actually stored. The index structure should be the same size that the 2D regular grid defined by the SBRT ($width \times height$); hence, the related stack is obtained by projecting the xy coordinates of the ray position on this grid. This projection provides a pointer which

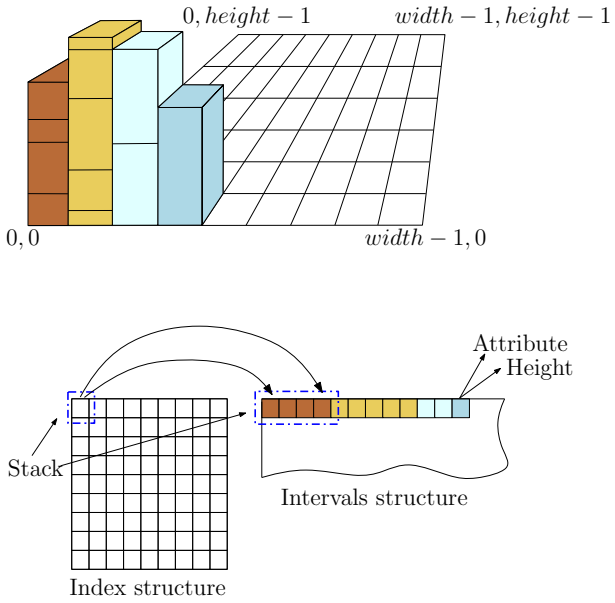

Figure 3: Overview of the memory layout (bottom) for a SBRT geomodel (top)

is used as an index for this index structure. Each element of the index structure encodes a 1D index (i.e., the pointer to the interval structure) and the number of intervals contained in the stack. The interval structure organizes each stack in a linear manner storing for each element, the interval attribute and its accumulated height. The interval attribute is represented by an integer number that is used as index for sampling a discrete transfer function, where the optical properties of the materials (such as color or opacity) are stored. This memory organization is shown in Figure 3.

For a more detailed description of this volume rendering procedure, we refer to [13].

Each rendered layer is added to the output framebuffer in sequence. Thus, for every layer a depth buffer has to be computed. This stores for each pixel the distance from the virtual camera to the object rasterized or hit by the raycast. When a new layer is rendered, their depth values must be compared with the previous one in order to write in the framebuffer only those non-occluded pixels.

Our visualization and analysis tool has been enriched with additional useful features, such as the attenuation of selected strata, visualization of cross sections, as well as the application of textures like the terrain orography or topographic maps. In case of stratum attenuation, a blending model has to be applied among the rendered layers.

Figures 5-8 show some visualization examples.

4.1 Camera navigation

Layers representing subsoil structures such as sewage or subway networks may be hidden if a field layer that represent the terrain surface is loaded into the system. In order to allow an inspection of these layers, besides the attenuation of the upper layers, we define a procedure based on the camera position.

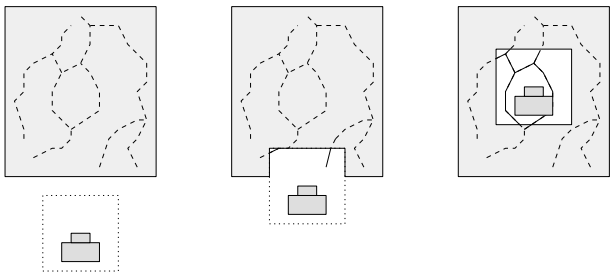

Figure 4: Description of the internal inspection of the data

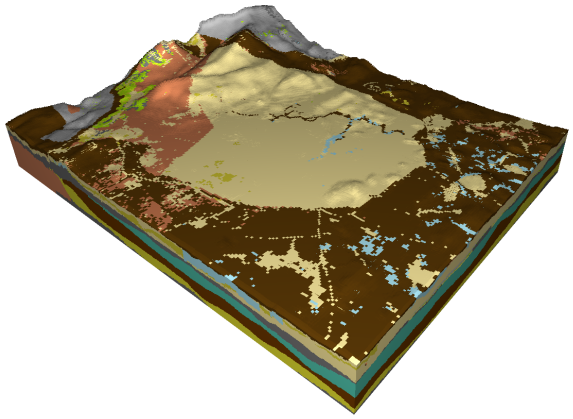

Figure 5: Example of visualization of a SBRT layer

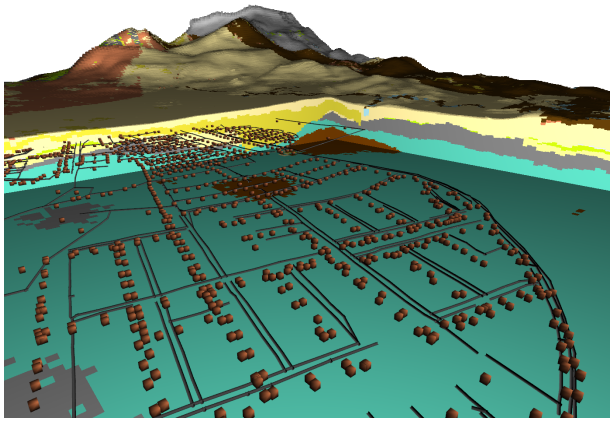

Figure 6: A close-up viewing of three different layers. Two subsurface vector layers are combined with a SBRT layer. The vector layers represent a sewage network (segment layer) and a set of manholes (point layer)

This procedure consists of enclosing the camera in a bounding volume, in this case an Axis Aligned Bounding Box (AABB), locating the camera at its center. Consequently, any part of the layer that collides with the AABB is excluded from the visualization. This feature allows moving the camera into the volumetric dataset in a

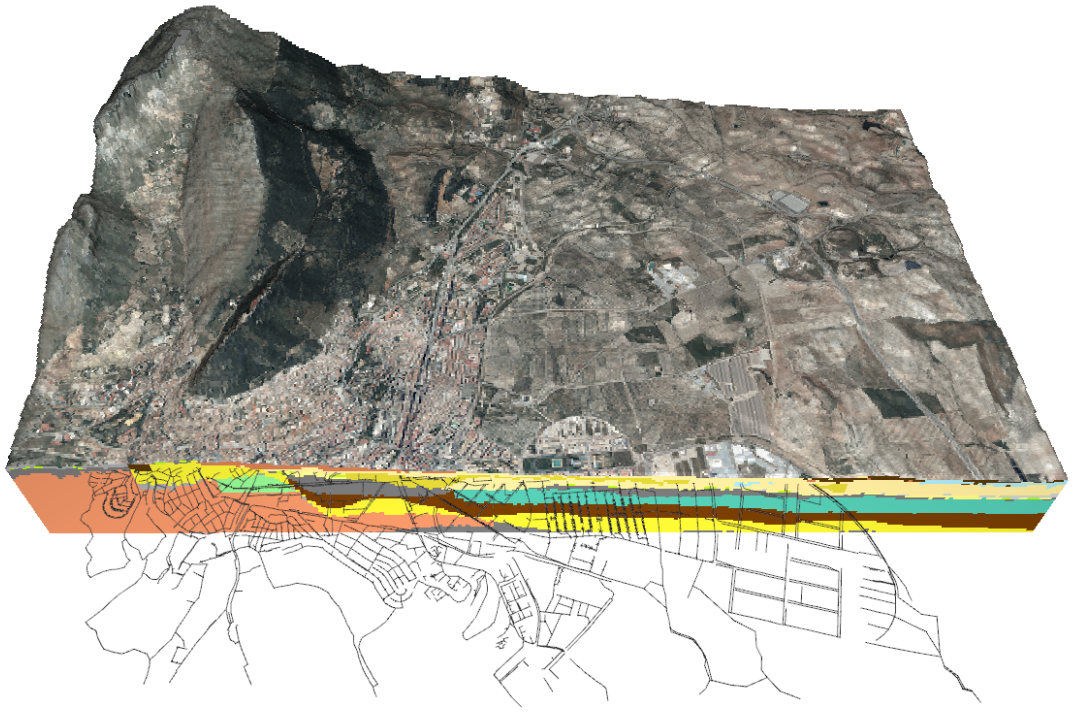

Figure 7: Cross sectioned field layer. The cut permits the visualization of a subsurface vector layer. In addition, an orthophoto is applied on the field layer surface

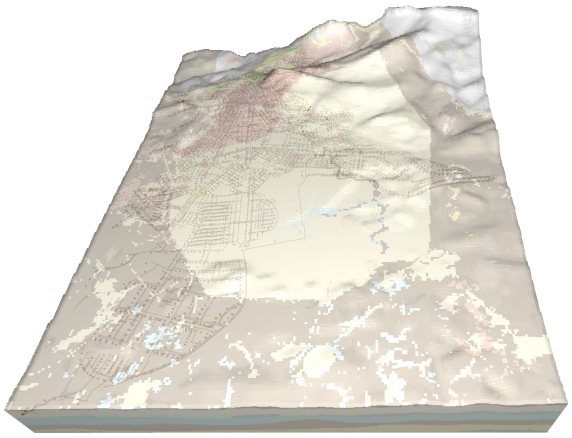

Figure 8: Example of attenuation of a field layer. A subsurface vector layer can be depicted

very natural way, showing hidden layers of materials and internal structures of the vector layers.

This is a very powerful visual tool for GIS professionals and geoscientists that allows, in a straightforward way, to inspect the relation between objects and materials, checking what materials are traversed by a pipeline, or how far a specific object is from a given material, for instance. Figure 4 illustrates this procedure.

4.2 Performance analysis

To test the visualization method above described, we have measured both the performance on rendering and the memory used by several datasets. In Dataset A, two layers have been loaded: a field layer modeling a SBRT which corresponds to a model of $200 \times 250 \times 320$ voxels and a network vector layer with 9153 segments. In Dataset B a SBRT layer which corresponds to a model of $400 \times 500 \times 400$ voxels and two vector layers, one with 6978 point objects and another with 9153 segment objects are loaded. Finally, Dataset C contains the same layers as Dataset B but with an increase of the SBRT size ($800 \times 1000 \times 800$ voxels). The SBRT datasets were obtained from the DINOloket database [17], whereas vector data were provided by a water management company.

The hardware setup for the experiments consisted of a PC equipped with an Intel Core i7-4790 CPU running at 3.60 GHz with 16 GB of RAM and an NVIDIA GeForce GTX 970 graphic card. The framework proposed has been implemented in C++ and OpenGL 4.5 as 3D graphic library.

Regarding the implementation of the SBRT structures, we used a new feature provided by OpenGL in its 4.3 version for storing large amount of data in the GPU, the Shader Storage Buffer Objects (SSBO) [39]. Therefore, two buffers are used for encoding each

structure. The buffer of indices is filled by an array of structures formed by two 32-bits unsigned integers (the pointer and the stack size), while the intervals buffer is composed by two 16-bits floating-point numbers (the attribute and the height of the intervals). The rendering method has been performed on a 1920×1080 viewport.

Each dataset was visualized in different camera positions inside and outside the volume provided by the field layer. As shown in Table 1, the visualization method achieved quite acceptable results, reaching at any time for all the datasets interactive frame rates.

5 OPERATIONS

In this section we describe a set of general hybrid operations between field and vector layers and their application to solve more specific problems in engineering and geospatial applications.

Our system can include operations among data from the same type as a regular GIS such as, the merging of two SBRT layers or unary operations like the test of a pipeline for monotonicity. Nevertheless, in this paper we focus on we called *hybrid operations* (i.e. operations that involve raster and vector data types).

GIS operations can be grouped into three main categories: access, analysis and visual operations. The latter has been already depicted in Section 4. Access operations have also been discussed above but in a unary scope. The data structures explained in the logical model enable efficient access to the spatial data both in field and vector layers. However, it may be necessary to identify what elements of a specific layer are covered by the selection performed in another layer. For example, given a selected stratum in a field layer, it is interesting to know if a pipe or a well contained in a vector layer lies within the selection. The opposite case is also relevant. Given a subsurface network layer, the total amount of materials that the segments traverse can be retrieved. This simple operation can be a powerful tool for cost estimation and planning.

5.1 Access operations

In this subsection, a definition of the basic hybrid access operations of our system is described.

Using the approach introduced in this paper, the extraction of a vector feature included within a selected stratum can be addressed in a straightforward way. Due to the rectangular shape of the cell grid of a SBRT, every stack, as well as each interval contained on it, can be defined by a cuboid. Therefore, given the set of intervals selected, the selection of the vector objects is carried out by performing a collision detection of them with the octree nodes (actually cubes) that organizes the vector layer. This box-box collision detection procedure is one of the simplest and fastest in the field of Computational Geometry [37]. Finally, the operation referred as vector-from-field access, returns in a new layer those vector entities contained in this collision region. For each selected interval, the complexity of the operation is $O(log_8(m)) + O(k)$, being m the maximum level of the tree and k the maximum number of elements, in this case vector elements, in a leaf node.

On the other hand, in order to obtain the SBRT intervals included in those geometric entities selected in a vector layer, a different approach must be followed depending on the type of this vector layer. In the case of a vector layer containing point features, the operation (called field-from-vector access) is as simple as sampling the SBRT on the selected points. Thus, a new layer with the sampled intervals is created. If the vector layer represents segment or network features, a cell walking algorithm can be applied in order to determine the traversed stacks [4]. This algorithm selects those cells in a grid intersected by a segment given its two extreme points, similar to that commonly used in line rasterization. If the vector layer contains more than one segment, the algorithm must be repeated for every object. Then, with a line-box collision detection procedure between the stack and the segment that it traverses, the exact interval is obtained. Finally, in the case of polygonal or solid vector layers, the lower and upper coordinates ($p_{min} = (x_{min}, y_{min}, z_{min}), p_{max} = (x_{max}, y_{max}, z_{max})$) from all the points are selected in order to generate the AABB of the objects of interest. The collision methods are just carried out on the stacks included in this AABB. Therefore, a polygon-box collision method for the polygonal case and a polyhedral-box collision for the solid case has to be performed [37] The complexity order of this query is $O(1) + O(n)$ being n the number of selected vector elements.

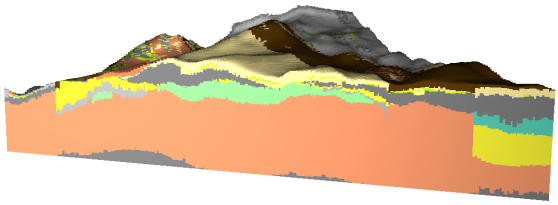

Figure 9: Cross-section allowing a terrain profile display

5.2 Case studies

The access operations defined above serves as a basis for more specific analysis solutions. In this subsection we present three different common problems in subsurface engineering surveys, in which the analysis of infrastructures such as tunnels or pipelines must be carried out in conjunction with their geological environment. Also we present a solution provided for our system.

- **Terrain profile and altitude**. In tunneling and underground excavation, there are required techniques capable of extracting the terrain longitudinal profile [34]. This feature provides engineers with clear, fast and visual information about the characteristics of the terrain, such as its material distribution as well as its maximum and minimum height. Let us consider an engineering survey for the planning of a tunneling work. Two layers have to be defined in the system, one SBRT field layer and a second vector layer representing a tunnel segment. By performing a field-from-vector access operation, a new field layer with the set of stacks located along the segment can be generated. Figure 9 shows the visual result of this operation. As can be seen, the back intervals of the profile are also included in the new layer.
- **Trench excavation**. In the process of maintenance of sewage systems is very usual to replace certain parts of the system [29]. It is carried out by digging trenches in the right places

Table 1: Performance results

	Memory usage (MB)		Average frame rate (fps)	
	Vector data (total objects)	Field data	Exterior	Interior
Dataset A	1.67 (9153)	1.56	132	245
Dataset B	2.31 (16131)	6.19	106	178
Dataset C	2.31 (16131)	29.06	79	108

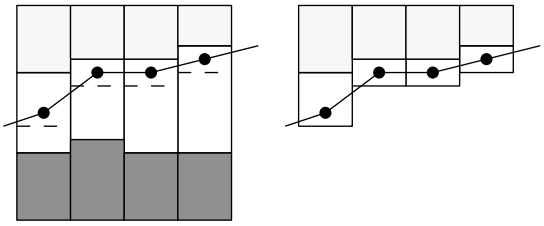

Figure 10: Procedure of trench excavation. Field data located above the vector objects are selected

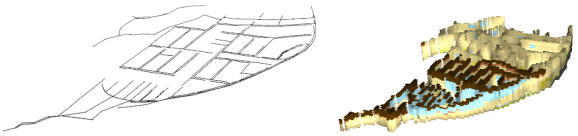

Figure 11: Result of applying a trench extraction operation. The stacks shown in the right side belong to the output layer

in order to replace the old pipe with a new one. Here again, a field-from-vector access have to be computed. The operation retrieves a new SBRT layer with all those stacks containing the selected segments of a network vector layer. Then, for each stack, it should be deleted the part which is below the z coordinate of the segment objects (Figure 10). The new layer can provide information about the amount and percentage of each material that is necessary to remove. Figure 11 shows an example of this operation.

- **Buffer operation**. One of the operations present in most GIS packages is buffering. This tool creates an area of influence around a particular object in order to perform proximity analysis. For example, this operation can indicate whether a point feature is at least at a certain distance from a specific material. Once the stratum is selected, a vector-from-field access operation can be applied in order to test the inclusion of the buffer (a sphere for a 3D point) within the stratum.

To conclude this section, we want to highlight that a implementation of our framework can benefit from a CPU multicore or GPU implementation. The SBRT is well suited for a parallel implementation of many algorithms. In fact this feature was used in our tool using the GPU massive parallelism for executing the raycasting visualization algorithm in realtime.

6 CONCLUDING REMARKS

In this paper, we have presented a framework for the hybrid visualization of 3D spatial data. This system allows the possibility of navigating freely through the hybrid data, using an efficient rendering. In addition, we have enhanced the system with some operations that require the use of field and vector layers. This framework maintains in every moment the nature of each layer both for visualization and operations purposes. Therefore, no conversion methods between the data are required, avoiding any loss of resolution. Unlike the usual methods given by current commercial and open source GIS, this approach presents an integrated framework to manage visualize and perform operations with heterogeneous spatial data that are central aspects in the development of Smart Cities. We use a very compact representation for the field data both in main memory and GPU, namely Stack-Based representation of terrains. This representation is appropriate for volumetric terrain models because the data are structured following a layer-cake model.

Nevertheless, the framework can be improved and extended in many ways. Although we have demonstrate that keeping the nature of the different data types is an advantageous property, it is interesting to go one step further and study the use of a general representation and its implementation for field and vector types. Finally, we have planned to integrate the described framework into a complete open source hybrid 3D GIS package.

ACKNOWLEDGMENTS

This work has been partially funded by the Ministerio De Economía y Competitividad of Spain under the I+D+i research project TIN2014-58218-R, and by the University of Jaén through the predoctoral research grant Acción 15.

REFERENCES

[1] E. Arnone, A. Francipane, A. Scarbaci, C. Puglisi, and L. V. Noto. 2016. Effect of raster resolution and polygon-conversion algorithm on landslide susceptibility mapping. *Environmental Modelling and Software* 84 (2016), 467–481. https://doi.org/10.1016/j.envsoft.2016.07.016

[2] B. Benes and R. Forsbach. 2001. Layered data representation for visual simulation of terrain erosion. In *Proceedings Spring Conference on Computer Graphics*. https://doi.org/10.1109/SCCG.2001.945341

[3] Filip Biljecki, Hugo Ledoux, and Jantien Stoter. 2017. Generating 3D city models without elevation data. *Computers, Environment and Urban Systems* 64 (2017), 1–18. https://doi.org/10.1016/j.compenvurbsys.2017.01.001

[4] J. E. Bresenham. 1965. Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4, 1 (1965), 25–30. https://doi.org/10.1147/sj.41.0025

[5] Stephen Brooks and Jacqueline L. Whalley. 2008. Multilayer hybrid visualizations to support 3D GIS. *Computers, Environment and Urban Systems* 32, 4 (jul 2008), 278–292. https://doi.org/10.1016/j.compenvurbsys.2007.11.001

[6] Tao Chen and Markus Schneider. 2009. Data Structures and Intersection Algorithms for 3D Spatial Data Types Categories and Subject Descriptors. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 148–157.

[7] Yujiao Chen, Holly W. Samuelson, and Zheming Tong. 2016. Integrated design workflow and a new tool for urban rainwater management. *Journal of Environmental Management* 180 (2016), 45–51. https://doi.org/10.1016/j.jenvman.2016.04.059

[8] Lidija Čomić, Leila De Floriani, Federico Iuricich, and Ulderico Fugacci. 2014. Topological modifications and hierarchical representation of cell complexes in arbitrary dimensions. *Computer Vision and Image Understanding* 121 (apr 2014), 2–12. https://doi.org/10.1016/j.cviu.2013.11.011

[9] ESRI. Environmental Systems Research Institute. 2017. ArcGIS. http://www.esri.com/arcgis/about-arcgis. (2017).

[10] Nivan Ferreira, Marcos Lage, Harish Doraiswamy, Huy Vo, Luc Wilson, Heidi Werner, Muchan Park, and Claudio Silva. 2015. Urbane: A 3D framework to support data driven decision making in urban development. *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)* 00 (2015), 97–104. https://doi.org/doi.ieeecomputersociety.org/10.1109/VAST.2015.7347636

[11] Geosoft Inc. 2017. *Target for ArcGIS*. http://www.geosoft.com/products/arcgis-extensions/target-arcgis/overview

[12] Michael F. Goodchild, May Yuan, and Thomas J. Cova. 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science* 21, 3 (2007), 239–260. https://doi.org/10.1080/13658810600965271

[13] Alejandro Graciano-Segura, Antonio Jesús Rueda-Ruiz, and Francisco Ramón Feito-Higueruela. 2017. Real-time visualization of 3D terrains and subsurface geological structures. *Advances in Engineering Software (In press)*.

[14] GRASS Development Team. 2016. *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.0*. Open Source Geospatial Foundation. http://grass.osgeo.org

[15] Gerhard Gröger and Lutz Plümer. 2012. CityGML âĂŞ Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71 (jul 2012), 12–33. https://doi.org/10.1016/j.isprsjprs.2012.04.004

[16] Gerhard Gröger and Lutz Plümer. 2012. Transaction rules for updating surfaces in 3D GIS. *ISPRS Journal of Photogrammetry and Remote Sensing* 69 (apr 2012), 134–145. https://doi.org/10.1016/j.isprsjprs.2012.03.004

[17] J. L. Gunnink, D. Maljers, S. F. Van Gessel, A. Menkovic, and H. J. Hummelman. 2013. Digital Geological Model (DGM): A 3D raster model of the subsurface of the Netherlands. *Geologie en Mijnbouw/Netherlands Journal of Geosciences* 92, 1 (2013), 33–46. https://doi.org/10.1017/S0016774600000263

[18] Jiateng Guo, Lixin Wu, Wenhui Zhou, Jizhou Jiang, and Chaoling Li. 2016. Towards Automatic and Topologically Consistent 3D Regional Geological Modeling from Boundaries and Attitudes. *ISPRS International Journal of Geo-Information* 5, 2 (2016), 17. https://doi.org/10.3390/ijgi5020017

[19] Markus Hadwiger, Joe M. Kniss, Christof Rezk-salama, Daniel Weiskopf, and Klaus Engel. 2006. *Real-time Volume Graphics*. A. K. Peters, Ltd. 497 pages.

[20] Haibo Hu. 2014. An algorithm for converting weather radar data into GIS polygons and its application in severe weather warning systems. *International Journal of Geographical Information Science* 28, 9 (2014), 1–16. https://doi.org/10.1080/13658816.2014.898767

[21] Flemming Jørgensen, Rasmus Rønde Møller, Lars Nebel, Niels-Peter Jensen, Anders Vest Christiansen, and Peter B. E. Sandersen. 2013. A method for cognitive 3D geological voxel modelling of AEM data. *Bulletin of Engineering Geology and the Environment* 72, 3-4 (2013), 421–432. https://doi.org/10.1007/s10064-013-0487-2

[22] Oliver Kersting. 2002. Interactive 3D Visualization of Vector Data in GIS. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*. McLean, Virginia, USA, 107–112.

[23] Matthias Kreuseler. 2000. Visualization of geographically related multidimensional data in virtual 3D scenes. *Computers & Geosciences* 26 (2000), 101–108.

[24] Y. Liu, M. F. Goodchild, Q. Guo, Y. Tian, and L. Wu. 2008. Towards a General Field model and its order in GIS. *International Journal of Geographical Information Science* 22, 6 (2008), 623–643. https://doi.org/10.1080/13658810701587727

[25] Paul A Longley, Michael F Goodchild, David J Maguire, and David W Rhind. 2016. *Geographic Information Science and Systems* (4 ed.). John Wiley & Sons.

[26] Michael Maxelon, Philippe Renard, Gabriel Courrioux, Martin Brändli, and Neil Mancktelow. 2009. A workflow to facilitate three-dimensional geometrical modelling of complex poly-deformed geological units. *Computers & Geosciences* 35, 3 (mar 2009), 644–658. https://doi.org/10.1016/j.cageo.2008.06.005

[27] Martien Molenaar. 1990. A formal data structure for 3D vector maps. In *Proceedings of EGIS'90 (GI '10)*. Amsterdam, The Netherlands, 770–781.

[28] Mohammed Mostefa, Leila De Floriani, and Paola Magillo. 2009. Morphology Analysis of 3D Scalar Fields Based on Morse Theory and Discrete Distortion. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Seattle, Washington, 187–196.

[29] Department of Environmental Quality. State of Idaho. *Individual and Subsurface Sewage Disposal Systems State. Technical Guidance Manual Individual*.

[30] T. Ortner, J. Sorger, H. Steinlechner, G. Hesina, H. Piringer, and E. GrÃűller. 2017. Vis-A-Ware: Integrating Spatial and Non-Spatial Visualization for Visibility-Aware Urban Planning. *IEEE Transactions on Visualization and Computer Graphics* 23, 2 (Feb 2017), 1139–1151. https://doi.org/10.1109/TVCG.2016.2520920

[31] F. Penninga and P. J. M. Van Oosterom. 2008. A simplicial complex-based DBMS approach to 3D topographic data modelling. *International Journal of Geographical Information Science* 22, 7 (2008), 751–779. https://doi.org/10.1080/13658810701673535

[32] Morakot Pilouk. 1996. *Integrating Modelling for 3D GIS*. Ph.D. Dissertation. ITC Netherlands.

[33] Pitney Bowes Inc. 2017. MapInfo Discover. GIS for the Geosciences. http://www.pitneybowes.com/us/location-intelligence/geographic-information-systems/mapinfo-discover-3d.html. (2017).

[34] Pierre-jean Pompée. 2004. Channel Tunnel Rail Link, planning, design and associated urban development. *Tunnelling and Underground Space Technology* 19, 4-5 (2004), 351. https://doi.org/10.1016/j.tust.2004.01.044

[35] Oriol Pueyo and Gustavo Patow. 2014. Structuring urban data. *The Visual Computer: International Journal of Computer Graphics* 30, 2 (2014), 159–172. https://doi.org/10.1007/s00371-013-0791-7

[36] QGIS Development Team. 2017. QGIS A Free and Open Source Geographic Information System. http://www.qgis.org/en/site/index.html. (2017).

[37] Philip J. Schneider and David Eberly. 2002. *Geometric Tools for Computer Graphics*. Elsevier Science Inc., New York, NY, USA.

[38] Andrea Scianna. 2013. Building 3D GIS data models using open source software. *Applied Geomatics* 5, 2 (2013), 119–132. https://doi.org/10.1007/s12518-013-0099-3

[39] G. Sellers, R. Wright Jr, and Haemel N. 2016. *OpenGL SuperBible. Comprehensive Tutorial and Reference* (7 ed.). Addison Wesley.

[40] Jiangfeng She, Yang Zhou, Xin Tan, Xingong Li, and Xingchen Guo. 2017. A parallelized screen-based method for rendering polylines and polygons on terrain surfaces. *Computers and Geosciences* 99, January 2016 (2017), 19–27. https://doi.org/10.1016/j.cageo.2016.10.011

[41] Dayong Shen, David W. Wong, Fernando Camelli, and Yuling Liu. 2013. An ArcScene plug-in for volumetric data conversion, modeling and spatial analysis. *Computers & Geosciences* 61 (dec 2013), 104–115. https://doi.org/10.1016/j.cageo.2013.08.004

[42] W. Tegtmeier, S. Zlatanova, P.J.M. van Oosterom, and H.R.G.K. Hack. 2014. 3D-GEM: Geo-technical extension towards an integrated 3D information model for infrastructural development. *Computers & Geosciences* 64 (mar 2014), 126–135. https://doi.org/10.1016/j.cageo.2013.11.003

[43] Matthias Thöny and Renato Pajarola. 2015. Vector Map Constrained Path Bundling in 3D Environments. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS '15)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/2833165.2833168

[44] Yong Tian, Yi Zheng, and Chunmiao Zheng. 2016. Development of a visualization tool for integrated surface water âĂŞ groundwater modeling. *Computers and Geosciences* 86 (2016), 1–14. https://doi.org/10.1016/j.cageo.2015.09.019

[45] Roman Trubka, Stephen Glackin, Oliver Lade, and Chris Pettit. 2016. A web-based 3D visualisation and assessment system for urban precinct scenario modelling. *ISPRS Journal of Photogrammetry and Remote Sensing* 117 (2016), 175–186. https://doi.org/10.1016/j.isprsjprs.2015.12.003

[46] Linda van den Brink, Jantien Stoter, and Sisi Zlatanova. 2013. UML-Based Approach to Developing a CityGML Application Domain Extension. *Transactions in GIS* 17, 6 (2013), 920–942. https://doi.org/10.1111/tgis.12026

[47] G.H. Weber, S.E. Dillard, H. Carr, V. Pascucci, and B. Hamann. 2007. Topology-Controlled Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 330–341. https://doi.org/10.1109/TVCG.2007.47

[48] Linwang Yuan, Zhaoyuan Yu, Wen Luo, Lin Yi, and Guonian Lü. 2014. Multidimensional-unified topological relations computation: a hierarchical geometric algebra-based approach. *International Journal of Geographical Information Science* 28, 12 (jun 2014), 2435–2455. https://doi.org/10.1080/13658816.2014.929136

[49] Xiaolei Zhang, Yong Han, Dongsheng Hao, and Zhihan Lv. 2016. ARGIS-based outdoor underground pipeline information system q. *Journal of Visual Communication and Image Representation* 40 (2016), 779–790. https://doi.org/10.1016/j.jvcir.2016.07.011

[50] S. Zlatanova, J. Beetz, A.J. Boersma, A. Mulder, and J. Goos. 2013. 3D Spatial Information Infrastructure for the Port of Rotterdam. (apr 2013).