IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 9, 2020, accepted March 19, 2020, date of publication March 30, 2020, date of current version April 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984035

Topological Data Models for Virtual Management of Hidden Facilities Through Digital Reality

LIDIA M. ORTEGA, JUAN M. JURADO, JOSÉ LUIS LÓPEZ RUIZ, AND FRANCISCO R. FEITO
Universidad de Jaén, 23071 Jaen, Spain

Corresponding author: Lidia M. Ortega (lidia@ujaen.es)

This work was supported in part by the Ministerio de Economía y Competitividad, and in part by the European Union (via ERDF funds) through the Research Project under Grant TIN2017-84968-R.

ABSTRACT The integral management of facilities located in the subsoil or hidden in walls and ceilings entails numerous challenges in terms of accurate positioning and data update. We describe the characteristics of the information system designed to act in the life cycle of facilities networks. We focus on the data model design, whose core connects topologically the 3D geometric objects representing facilities components. This scheme enables CRUD (Create, Read, Update, Delete) operations directly on the 3D objects of the digital reality, and through desktop or mobile devices. The system is implemented in a client-server architecture. The server maintains the complete topological data model in a spatial database. This model is redesigned and replicated in the memory of any client device with graphical capabilities. Conventional computers can enable Virtual Reality, while mobile devices can access ubiquitously surrounding facilities through Augmented Reality. In addition, the design of a third data model allows the exchange of this information using a standard format such as CityGML. Current facilities networks have been used to test the CRUD operations functionality.

INDEX TERMS CRUD operations, facilities management, augmented reality, virtual reality, CityGML, underground infrastructures, 3D-GIS, BIM, ARCore.

I. INTRODUCTION AND STATE OF THE ART

Many of the facility networks that make cities and buildings work are hidden. The infrastructure under the surface preserves the available space and provide security and isolation. This happens in most installations that transport liquids, gases, electricity or data. A proper planning of these infrastructure allows to reduce the environmental impact while providing better service for citizens [1]. Further to the design, their maintenance is critical, involving different companies and public administrations. In order to facilitate the optimal management of energy, transport and supplies, the Information and Communications Technology (ICT) plays an increasingly decisive role [2]. These objectives are considered nowadays within both the so-called Smart Cities and Facilities Management (FM).

In general terms, difficulties associated to facility network management are very diverse [3]. Principally, they coexist in very close spaces and they are not directly visible because of being installed behind walls, ceilings, pavement or asphalt. This may cause multiple problems in the management process, for instance the lack of accuracy in positioning when digging the surface to access these infrastructure elements [4]. Further to the design, there can be different companies and public institutions involved along their lifecycle. The ubiquitous nature of maintenance tasks is not facilitated by means of Information Systems (IS) accessible only from the office. It is still common to use paper plans in the workplace. Although this format is universal and very transportable, it is not the most suitable for real-time data updates. This can lead to numerous inconsistencies, especially in response to the geometric and spatial nature of facility networks [5]. As a consequence, the information system supporting infastructure networks should be able to facilitate all these management tasks. This implies enabling the set of CRUD (Create-Read-Update-Delete) operations on the information from any position and by anyone involved.

The design of new infrastructure has been traditionally carried out using Computer-Aided Design (CAD) tools, and more recently by Building Information Modeling (BIM).

The associate editor coordinating the review of this manuscript and approving it for publication was Sohail Jabbar.

What they both have in common is the 3D modeling applied to many different areas such as Architecture, Engineering and Construction (AEC). However, BIM goes beyond design including richer product information and allowing engineers and contractors work collaboratively on the same project reducing time and errors. Accordingly, BIM facilitates the construction workflow [6]. Even though BIM are mainly software tools for design and construction phases, they are perceived as helpful during the whole lifecycle [7], [8]. However in this regard, they still must evolve to enhance portability, accessibility and real time manipulation for all actors involved [9].

So far, the Geographic Information System (GIS) has been considered more appropriate to manage spatial information since they maintain databases with specific geometric types while allowing analysis, visualization and decision support. In reality, most GISs still work in 2D. Therefore, the incursion of the third dimension in a so-called 3D GIS, is considered a real contribution in the field of subsurface management [10], allowing data visualization and analysis [11]. Evidently, a 3D GIS is not a simple extension of the two-dimensional version. The information cannot be included as an isolated set of 3D models. The lack of topology that typically affects 2D geometric layers should be avoided in 3D GISs [12]. The problem of current 3D GISs, both commercial and open-source, is that they are not specific tools for facilities. They fail in accurate visualization, specific analysis, usability or ubiquity. Ultimately, there is a tendency to consider the merger between BIM and 3D GIS to benefit from both design and spatial data, not only during planning and construction but throughout the system's life cycle [13], [14].

Beyond the system used to support the infrastructure, the data model used to facilitate the CRUD operations must maintain topological relations. Topology defines the ways in which nodes or elements of a network are associated or interconnected [15]. Their importance in the field of geosciences has been assumed by the Open Geospatial Consortium (OGC), the international organization in charge of providing standard technical documents that detail interfaces or encodings. Topological connectivity of features is included in their standard data-formats and database definitions [16], [17]. In general terms, spatial databases include these operators to establish and even calculate the spatial relationship between the geometric elements [18]–[20]. BIM also sees this need defining topological operators for defining efficient processing of spatial queries [21].

In the specific case of facilities connected as a network, not all topological operations are needed. The type of network topology depends mainly on whether it is wiring or pipes, and the nature of the transported element. One of the most widely used in water or gas supply networks is *Multipoint*, in which a collection of nodes is connected point-to-point [22]. In order to define how two facility elements are joined, the *Meet operator* may be sufficient, although in practice it is known that facilities must overlap to establish this connection [23]. In short, both for its construction and for its subsequent maintenance, the infrastructure networks need topological connections and semantics to be feasible [24], for instance to study the behavior of water distribution systems [25], or for an efficient data access in BIM [21], [26].

In addition to spatial databases, on numerous occasions a memory data model is needed to expedite data manipulation on CRUD operations. This is of importance when only a portion of the scene is being navigated, for example, when an operator is making maintenance tasks. representation on the device memory. Graphs are mathematical structures linking nodes and edges that provide topology inherently, which makes it valid for representation of facility networks [25], [27]. Similarly, topology is given by definition in combinatorial maps, which is used to define topological representations [28].

Whether BIM or 3D GISs are used to design, built or maintain facilities, another of the underlying problems is the way of operating directly with them. Except for the design, the rest of phases are carried out at the construction site [29]. Then a repetitive process begins in which technicians visualize the project, interpret it and act accordingly to the technical specifications. The format in which the project is provided for its management in a ubiquitous manner does not correspond with the 3D models used in the design. They are normally converted to 2D plans with a high level of abstraction that makes data understandable only to qualified personnel. Additional maintenance phases uses the same data format.

Mobile devices offer a new paradigm to the way technicians work in the AEC areas. However as stated above, any Digital Reality (DR) enriched with geometric and thematic data requires a spatial information system behind. This means that even though an information system is hosted in the server, a mobile device can access the surrounding virtual 3D representation of cables or pipes. Therefore, the mobile operates as the client in a client-server system [30].

The possibility of offering a 3D graphical interface on current mobile devices is possible today with the advancement of hardware and graphic libraries. Advances regarding urban infrastructures have focused mainly on virtual visualization in order to eliminate in some way the layer of material that makes them invisible. They have worked on numerous techniques based on computer vision [31], Virtual Reality (VR) and over all Augmented Reality (AR) [29], [32], [33]. The advantage of AR is also the desired ubiquity in these cases taking advantage of advances in Global Navigation Satellite Systems (GNSS) [34], [35]. Related to AR and VR, Mixed Augmented Reality (MAR) allows the user to be immersed in the real-world environment with additional digital content. These virtual objects can be used not only for visualization but for interaction [35]. In order to provide the corresponding spatial coherence to the two worlds, real and virtual objects must coincide spatially. That is, the virtual representation of an object must have the exact position as if it were a real object [36]. This matching process is the key to making it possible.

Even when the first operation with the hidden facilities in situ is a 3D visualization, the goal must go further, allowing interaction with the virtual models [37]. However, most approaches lack the mechanisms that provide information feedback, and neither the set of CRUD operations. In facility networks, CRUD operations means that if a well must be displaced a few meters in the virtual model, then the rest of pipes connected with it also change their relative positions. Thus, the information system must be able to maintain geometric and topographic restrictions when the 3D models are edited. This process makes sense during the construction or maintaining phases. Collisions with other facility elements must be considered, and the network must be consistent at the functional and structural level.

Additionally, the standardization of models and data formats are necessary for data exchange of urban facilities. The use of standards allows software developers to build open interfaces and encodings in their products and services. In the urban scope, the OGC provides the open data model CityGML, an XML-based (eXtensible Markup Language) format for the storage and exchange of virtual 3D city models [38], [39]. CityGML represents thematic and semantic 3D city models. Most thematic models found in an urban environment are covered: buildings, relief, transportation, land-use, tunnels, bridges, vegetation, water bodies and city furniture. The objects which are not yet specifically modeled in CityGML can be represented using generic objects and attributes, although CityGML also has the concept of ADE (Application Domain Extension) to extend the schema with new classes and attributes and defining an extra XSD (XML Schema Definition) file with its own namespace.

The particular case of urban facilities can be represented by the UtilityNetworkADE [40] which defines a topological network model for analyses and simulations on utility networks and supplying infrastructures. The model allows the representation of network components as 3D topographic city objects. The initial core definition is extended to cover both the topological and topographic representation as well as the functional and semantic classification of network objects [41]. It differentiates pipes, cables and canals as distribution elements of a network, and also those devices required for control, storage or measurement. This data model has been used in several applications such as energy analysis [42] or more recently in storm drainage network [43].

In this paper, we present the topological data model that provides CRUD operations on network facilities through the Graphical User Interface (GUI) of any client device in a system with a client-server architecture. The paper is organized as follows: Section II describes main objectives of the system designed with the topological model proposed in the paper. The specific instances of data models proposed for the database, memory and CityGML are described in Section IV. Section V shows different system applications to evince the potentiality of the topological model, for example for navigating through the scene or modifying the structure of the subsurface network through geometric editing tools. In this section, a check is also made on the connectivity of the network. Finally, Section VI provides conclusions and future work.

II. OBJECTIVES AND SYSTEMS OVERVIEW

According to the Introduction section, there are numerous desirable capabilities that an information system for infrastructure networks must have. This IS must be collaborative, manage spatial and thematic information, and even facilitate CRUD operations ubiquitously through a graphical interface. For that purpose, we design the core of our data model with all the geometric entities topologically connected. This allows to maintain geometric and topographic restrictions when editing the network structure. Assuming that maintenance tasks on the infrastructures are performed in situ, mobile devices are very useful tools to carry them out. For that purpose, the data model originally designed for a spatial database must be adapted to the main memory of a computer. This speeds up the process by managing only the portion of information that surrounds the worker. But more importantly, the replication and adaptation of this model to memory allows direct manipulation through a graphical interface using virtual o augmented reality. All changes performed in the model must be updated in the server via Internet, which allows feedback between the model and the three-dimensional view. Additionally, the system should be able to be exported using an standard exchange model. We design an extension of CityGML by redefining and simplifying the UtilityNetworkADE [40].

In summary, we describe the functionality and capabilities of our approach:

- **Spatial nature**: all the information related to the territory has a strong spatial character. The tools capable of handling this information are 3D GIS and BIM. We define our spatial data using an open-source spatial database with some of the capabilities associated to these software tools. This design is redefined in data structures to be efficiently managed in memory.
- **CRUD operations through a GUI**: Traditional data management functions are Created, Read, Updated or Deleted CRUD) through forms in text mode. However, the spatial nature of these infrastructures and the fact of being located under the surface makes these tasks more straightforward through Virtual or Augmented Reality techniques.
- **Intelligent and analytical**: The information about underground or subsurface elements must not only be viewed or consulted, it must also be used for decision-making and efficient management of the resources. This information is maintained not only as a collection of heterogeneous data, but through a topological data model, in which all elements are integrated and connected. This data model allows a large set of efficient analysis operations, which help all the operators involved to access the data and provide a joint decision.
- **Integrated, collaborative and on-line system**: The integration of spatial databases with standardized

protocols and formats is one of the main objectives of the Spatial Data Infrastructure (SDI). Accessibility of data through geo-portals has been standardized and their use and content have grown steadily. However, SDI is primarily used as an information repository, with some capabilities for visualization. The next challenge consists of accessing to these data through unified and integrated tools via web, for visualization and analysis, but also for a ubiquitous edition, if necessary. Several user profiles should provide different types of access to companies or entities with different associated potentiality. Public administrations and companies involved in the management and maintenance of the infrastructures could use an integrated management system, and even citizens could be able to provide information about failures or the level of satisfaction.

- **Client-server architecture**: The system should be accessible from any location in the city via the Internet. This requires a client-server architecture to allow any ubiquitous device to be connected. When considering that the editing process should also be possible at street level, this suggests that the data model must be prepared for being modified on a tablet or mobile device. Therefore, the information is not only retrieved from the server to the client for visualization, it must be transferred back to the server when the data is modified. Furthermore, mobile devices must be provided with protocols, interfaces and the appropriate data model to perform such operations.
- **Standardization**: Both data models described above, for database storage and memory access, are necessary for the system to work, although in an isolated way. In terms of data exchange, there are standard formats for urban data such as CityGML, an open data model and XML-based format for the storage and exchange of virtual 3D city models. The problem is that network-type facilities are not part of the CityGML definition. However, this scheme can be extended with the denominated ADE. This concept allows adding new classes and attributes which are not explicitly modeled in CityGML. In the particular case of utility networks, there already exists the Utility Network ADE. We use this extension as framework to carry out this exchange and standardization process.

III. GENERAL OPERATION SCHEME

In order to achieve the requirements described in Section II, we have designed a system with a classic client-server architecture as depicted in the scheme of Figure 1. The server runs on Ubuntu Linux and hosts a spatial database in PostGIS maintaining both geometric and thematic information of different facilities networks. PostGIS is a spatial database extender for PostgreSQL supporting geometric and geographic data types and their operations. The database, hosted in the server, maintains a set of registers of different nature determining a network of nodes and edges of a graph. As a matter of fact, it is irrelevant if the network transports liquids, gases or electricity. All the elements, pipes, wires, vaults or underpasses have the same representation at the geometric level, though not at the functional.

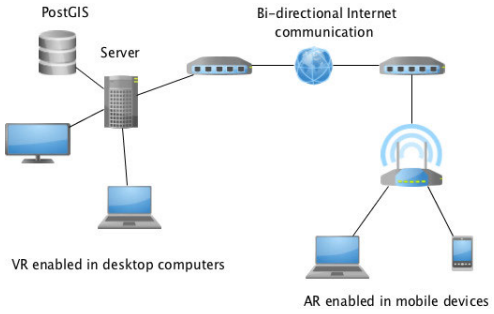


FIGURE 1. Clientserver architecture of the system.

The client side visualizes, interacts and also modifies the information when necessary. The advantages of this scheme are twofold: the client may be any device with certain graphics capabilities and can be placed in any site with an Internet connection. Thus, the system is capable of providing maintenance of infrastructure in situ, displaying, navigating and updating this information. The CRUD functions can be performed by editing registers, for instance if a pipe has a different material to the one given in the database. As is often the case, however, geometry is better accessed in situ through a graphical interface as described in Section V.

The original data given as input is provided by the different companies responsible for their management. Regarding outdoors infrastructure, we have tested the systems in charge of the sewage and the water supply. Additional data have been used indoors as electricity, water and network cabling. In all cases, the information is provided as files in DWG (from drawing) format. Each file contains several networks and each network is defined as a set of layers. In many cases, it is not easy to know which layers define a complete network. Overall coherence is only observed when viewing all the information together. Therefore, there is a lack of links and information connecting one layer with another. Data layers are either geometric or thematic. Geometry is given as a set of polygons or polylines, with UTM (Universal Transverse Mercator) coordinates for outdoor facilities. In these cases, depth information is not provided by the companies. Additional thematic data is sometimes associated to this geometry but their association is only perceived when displayed together on the screen. Therefore, all these layers with thematic and geometric information lack topology, either between elements of the same or of different layers.

In summary, the input to the data models has been pre-processed in order to create tables and classes from both geometric and thematic entities. Then, we create those topological relationships between elements from the same and different layers. Finally, they are provided with approximated heights according to the real DEM (Digital Elevation Model) of the city and considering the topographic constraints.

Similarly, a depth is set for floor and wall installations in building interiors.

Topology is computed when tables have been gathered in the database as isolated 2D points for node-type entities and polylines for lineal-type ones. The algorithm for connecting elements works independently for each type of network. The criterion for joining some components with others in related layers is proximity, which means that if two objects are close in the XY plane, it implies that there is a spatial connection. Data is mostly coherent in terms of X and Y coordinates, which facilitates this process. The proximity value chosen to establish when two elements are connected is a constant value or Epsilon. This premise works correctly in the majority of cases, however, when checking connectivity, some inconsistencies with the input data are discovered, as detailed in Section V.

The following premise, verified by the technicians, indicates that each section of pipe begins and ends in a node-type object. That implies that two sections of pipes can be joined directly but the intersection of three or more pipes is done by means of distribution elements such as manholes or sewer boxes in case of the sewerage system. This structure corresponds to the multipoint topology. Therefore, the solution is to find intersections between the geometric elements of the different layers adding a certain buffer of influence. Once these intersection points are obtained, these topological relations are added to the database according to the data model described in Section IV.

Proximity is also the criterion for associating thematic and geometric information. These layers of information indicate, for example, the type of material in a pipe. This info-type data is stored separately in the geometric core in the database, as described in the next section.

With regard to the approximate depth given to the infrastructure elements, we differentiate between indoors and outdoors. We need the Z value or depth information in order to provide a 3D data model. In the first case, the margin of error is small, since these elements are framed within the floors and walls, of known dimensions.

The sewerage and water supply networks are also provided with approximate values that can be easily adjusted later, once they are verified in reality. In order to make this process as accurate as possible, a digital elevation model (DEM) of the city area is used with five-meter precision.

For instance, sewerage is always under supply water network. Pavement gutters are known to be placed in the street flush with the asphalt. Their depth is no greater than one meter. The inspection covers of manholes are also flush with surface, and their depths can be up to three meters. Sewer boxes and driving pipes can be at different depths, usually between two and three meters. In most cases, the sewerage system works by gravity. The case of drinkable water is different since pressure valves and pumps determine its operating scheme. All these constrains are considered by an algorithm that tries to assign simulated values to the set of components. As is obvious under this assumption, these rough values should be checked later and updated in order to obtain the desired level of certainty. In many cases, this process is performed on site. This is the reason why additional techniques are needed to carry it out easily on site, as stated in Section V.

IV. DATA MODELS

In this section we describe the three data models designed with the aim of obtaining all the functionality described in Section II. The spatial database stores all information together with its topological relationships. However, this representation is not flexible enough to allow geometric editing on any client device. In the case of using mobile devices as a client, only a small bit of information will be needed. Therefore, this data model is re-designed in memory to allow direct geometry access. This allows the modification of each element or portion of a pipe or wire individually. Finally, in order to provide exchangeable models, we also provide the data model supporting the standard CityGML, the spatial data exchange issued by the OGC. In fact, these three representations are very similar and maintain the same topological structure. However, each of them is favored to provide the appropriate functionality.

The exact spatial location, the length of the conduits and the topological relations between different elements are considered to be the common geometric kernel of all of them. All these entities can be cataloged basically in two types: those associated to a point or a node in the space, and those related to a line. In the first case, we include manholes or wells for the sewerage system, or switches or routers in the case of a telecommunication network. All these elements can be located by a specific position in the space. Line entities are pipes or wires, whose geometric representation is given by a single line-segment or a polyline, depending on the input data. Therefore, there are two different representations of these entities at geometric level: nodes and edges as described in Graph Theory represented as points and segments or polylines respectively.

A. DATABASE DATA MODEL

The database system chosen for the sewerage information is PostGIS, the spatial database extender for PostgreSQL. PostGIS is free software and compatible with OGC standards. It provides full functionality to store and compute 2D and 3D geometric information.

After intersections are computed and the network of facilities is connected topologically, and estimated Z values are assigned to nodes, the database is ready to be used. As described above, all data models define a core with all the geometric entities of the facilities. Therefore, nodes and edges of all networks are stored in the same tables, regardless of their functionality. In fact, only distinction is made between indoor and outdoor infrastructure networks due to the coordinate system. This allows to represent each entity as unique in space under the premise that two elements of an infrastructure can be close, but they do not occupy the same space.

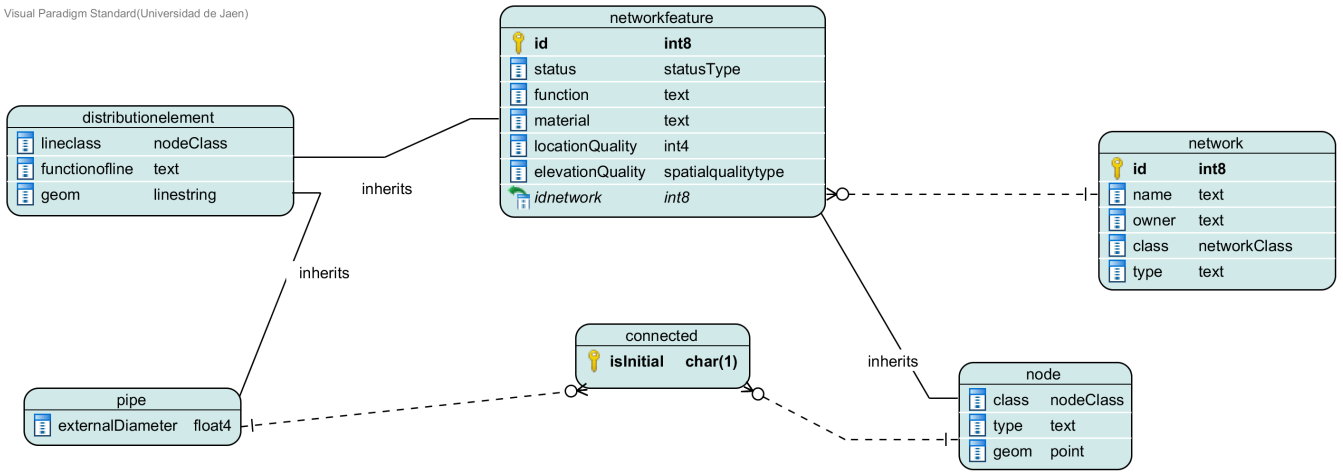


FIGURE 2. **Database design in PostGIS for the sewerage system.**

Thus, electricity, gas, water supply or even transportation are all included in the same tables at geometric level. Additional tables provide the rest of thematic information separately.

This common core provides the main advantage of computing geometric constrains in a simple and effective way. Topographic restrictions of the network can be checked on the same tables. For example, when trying to modify the depth of a sewer element, the electrical wiring or drinking water installation can be affected. The computation of these constraints is performed on the same database by using only one or two geometric tables instead of dozens, in case of complex models. Although it is true that, at this level, geometry is based on 3D points and polylines, the rest of their attributes such as thickness of pipes or the dimension of node-type elements are available in the database in different tables. In any case, the database is operated through SELECT operations with no need to perform costly JOIN operations.

Figure 2 shows the database design for any network based on pipes. Later we will see that it is also extensible for wiring. The structure defined by the UtilityNetworkAde [44] is reused as outlined below. In particular, some entities have the same nomenclature because they represent the same concept such as *Node*, *Network* and *Networkfeature*.

As observed in Figure 2, a *network* is defined a set of *networkfeature* elements. A network has a unique identification (id), a name and additional attributes for its identification. A *networkfeature* is any functional entity in this network, following the original design of the *UtilityNetworkADE*. This abstract class represents any of the elements of the infrastructure, both the node-type and the pipeline elements. It provides a unique identifier in the database and also gives information about the status, functionality and material of construction of any piece. The process of obtaining a position (X,Y,Z) is also stored, if it is real, calculated or interpolated. This is because the input data is not initially reliable regarding the depth of the elements, especially underground. Then, it provides information about the level of confidence in positioning each element, as well as the method with which it has been obtained. Finally, the attribute *idnetwork* identifies the network to which this element belongs. This means that the same table *networkfeature* maintains all the functional elements of all networks coexisting in the same place. Thus, sewerage, water supply and wires of telecommunication or electricity are all inserted in the same table but linked with their specific facility network.

PostgreSQL allows the inheritance relation between their entities, which facilitates modifying the traditional design of entities to make it hierarchical. This implies that a child table inherits all the attributes from its parent. A query can reference either all rows of a table and all of its descendant tables.

In [44], a *_Distributionelement* is an abstract class representing pipes, tunnels or cavities, introducing in this way all types of network components for distribution. In this paper, a *distributionElement* inherits from a *networkFeature* to represent any type of pipe, wire or even a canal as the means to transport some type of substance or signal. The class maintains the fields *lineClass* to define the type of functionality (unknown, MainLine, TransportLine, SupplyLine, HouseServiceLine). On the other hand, *functionioffline* determines the type of flow that transports. This table also maintains the geometry, the *linestring* data-type of PostGIS. It consists of a collection of tree coordinates for each node of the polyline, that is, a succession of type values (x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3...). Remember that some additional attributes are inherited from the table *networkfeature*.

A *pipe* is a specific case of *distributionElement*. Most of the fields of information about a pipe such as functionality or material are also necessary for any other type of network, so it is located in its ancestor classes. The only attribute added in the *pipe* table is related to its geometry, the width. The length is also inherited from the *distributionElement*. As can be predicted, this design supports adding more types of distribution elements, such as cables, canals or tunnels. All of them are

considered as child classes of *distributionElement* and incorporates additional and specific fields that characterizes them. For most wiring networks, the cross section is also important, but depending on the specific purpose, additional information should be also incorporated. For instance, the amperage in electrical networks.

In the same Figure 2, *node* is a child class of *networkfeature*. This entity represents any facility element defined as a point in the space. Wells, manholes, electrical receptacle boxes, control or measurement elements, or any element working as storage or union of several distribution elements is conceived as a node. The geometry is determined by a simple point in the space using native data types from PostGIS, which allows to take advantage of the geometric analysis capability of this spatial database. The specific *type* is identified as text (manhole, pavement gutter, etc.). Some other attributes are inherited from its parent class, such as the material of construction. The database does not add more information regarding behaviour. Algorithms associated to the system will be in charge of determining how the network behaves at the functional level.

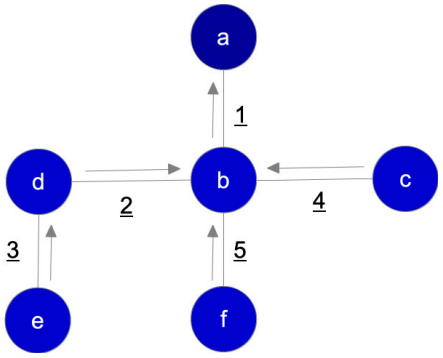
The last table in the design is *connected*, which provides the topological relation or link between nodes and pipes. In this implementation edges and nodes are connected bidirectionally. This relationship between a *Node* and an *Edge* follows the classic design multipoint topology, in which a pipe stars and ends in a node-type element such as a well or an electrical box. But it also symbolizes that a well or electrical box can be the recipient of several pavement gutters or wires, which implies that two or more conduits are connected to this node entity. This is in fact a many-to-many relationship in the database between nodes and polylines, which is normally implemented by means of a new table as link.

The original design of the *UtilityNetworkADE* described in [44] differs from the one proposed in this article in two fundamental aspects: (1) a simplification of the model, in order to be adapted to a compact database design and (2) a bi-directional topological model. This simplification consists of conceiving the whole network at the same level, that is, no subnetworks are contemplated. The *UtilityNetworkADE* is given as a recursive definition of subnets connected by means of *InterFeatureLink* elements. Different networks can also be linked by means of *NetworkLink* classes. These tables can be incorporated in the future if this hierarchy of levels is required. In any case, the core design allows to be extended but not modified in case of adding new feature networks.

The second difference is the way in which the topology is defined. In the initial design, an *Edge* is directly related with the type *Node* by means of the *start* and *end* relationships. However, the relationship from *Node* to *Edge* is not direct, they are connected through *FeatureGrapth*. As mentioned above, our approach recreates the mathematical structure of a graph by means of the table denoted as *connected*.

1) TOPOLOGY FOR NETWORK CONNECTIVITY

Full network connectivity can be checked directly on the spatial database or in the memory model, both representations are dual. In this section, we demonstrate the process to determine is the network stored in the database corresponds with a connected graph. In the particular case of sewerage, this network is known to be a graph with a single connected component. However, working with real data usually involves detecting inconsistencies in the input data. The process of constructing this topological model from the existing input data can fail if this information is incomplete or contains positioning errors. In these cases, the computational geometry algorithms that perform topology can generate erroneous results, that is, several connected components instead of a single one.



Networkfeature: k k = stack.pop() k is connected to:	x is connected to k x	x is visited	Stack if x is not visited then stack.push(x)
			{a}
node: a	pipe 1	no	{1}
pipe: 1	node a node b	yes no	{ } {b}
node: b	pipe 1 pipe 4 pipe 5 pipe 2	yes no no no	{} {4} {5,4} {2,5,4}
pipe: 2	node b node d	yes no	{5,4} {d,5,4}
node: d	pipe 2 pipe 3	yes no	{5,4} {3,5,4}
pipe: 3	node d node e	yes no	{5,4} {e,5,4}
node: e	pipe 3	yes	{5,4}
pipe: 5	node b node f	yes no	{4} {f,4}
node: f	pipe 5	yes	{4}
pipe: 4	node b node c	yes no	{} {c}
node c:	pipe 4	yes	{}

FIGURE 3. Network connectivity example.

The process of detecting these errors is done using the classic Depth-First Search (DFS) algorithm for traverse the graph representing the network infrastructure [45]. Figure 3 shows an example of network representing nodes and pipes, and the process followed for determining connectivity. The arrows are not representative at this level, they indicate the direction of water. Following the database design of Figure 2,

the class *connected* is in charge of performing the topological connection among distributionelement-type items. Both *node* and *distributionelement* types are *networkfeature* items considered as nodes in the graph. That is, pipes are also considered as graph nodes. Therefore, the many-to-many relationship between both types is obtained by means of this class *connected*.

The DFS algorithm uses an in-depth search strategy to traverse the entire graph by means of a stack. Once a node in the graph is visited, this is marked and never processed again. If the graph is fully connected, all elements in the graph are catalogued as visited. The process associated to Figure 3 starts in a node-type, the feature **a**. Any of them is valid to start with this DFS algorithm. The stack data structure is used to insert features waiting to be processed. While there are elements in the stack, the method continues by taking out the first element on the top, node **a** in the example. The feature **a** is connected in the topological database with one distribution element or pipe denoted as **1** through the class *connected*. This pipe is inserted in the stack, and processed immediately after. Its annexed element is the node-type **b**, that is also inserted in the stack. When **b** is taken out, the database through the class *connected* indicates that **b** is connected to three different pipes, **{2,5,4}**, which are inserted in the stack. Each of them is handled at a time and inserting in the stack its attached features, as the table indicates. In this case, the top of the stack, pipe **2**, is consulted and its attached node-type element **d** processed. When a feature has already been processed, it is marked as visited and never processed again. The method follows according to the table and finishes when the stack is empty. If after finishing the process there exits non-visited nodes in the graph, then these nodes can be detected and solved as described in Section V-C.

The same process can be used to check if the network works by gravity. According to the specifications given by the technicians, this is what happens in the sewerage network under study. However, the Z or depth values are most times nonexistent, which allows to define approximated heights following the network constrains.

B. MEMORY DATA MODEL

In previous sections, we refer to the need of designing a different data model in memory. Depending on this client device, all or only a portion of the model loaded in the database is replicated in memory. This model is in charge of facilitating the CRUD operations through the GUI. The spatial database maintained on the server is not always appropriate to be directly used on the client side. These devices can be used on the site, that is, directly on the infrastructures in the street or indoors, through an Internet connection. Data transmission can slow down the process preventing a real-time interaction. On the one hand, the client-side can be any mobile device with limited capabilities, regarding computing capacity or memory. However, they do not need to load the entire infrastructure set on the database, but the area in which they are working. Once this reduced area of geometric and thematic information is retrieved from the spatial database and charged in data structures in memory, the managing process through GUI operations is quick enough for real-time interaction.

Therefore, the main objective for adapting the database data model to memory is providing the infrastructure with speed and functionality, which means managing a complete set of CRUD operations as well as the graphic environment of any mobile device connected to the Internet. This may involve the creation of new geometry, when incomplete input data is observed or during the design phase of new facilities. The inaccuracy in the input data may imply the update of any infrastructure element. Finally, unusable or non-existent objects, which are normally located when performing an inspection in the subsoil, can also be eliminated on the client device.

The main modules of clases are shown in Figure 4. Each of them can be configured as a set of related classes. They follow the Model-View-Controller (MVC) software architectural design pattern. This allows to separate the internal representation from the way in which this information is presented to the user. In our particular case, we use different GUI technologies and devices, such as augmented reality enabled on mobile devices or virtual reality in desktop/laptop computers. Only the view and the controller need to be changed in this scheme system in order to adapt the visual results to the client device.

The set of classes denoted as *Controller* are in charge of the communication with the *View* and also with the *Model*. In fact, it is considered as the interface with the rest of entities in the system, also with the database through the module IO (In/Out). When the user interacts with the *View* for deleting or translate an object through the *GUI* package, this order is initially received by the controller. There are specific controller classes for AR and VR for positioning and camera control.

The *View* set of classes is in charge of visualizing the objects in the scene. It manages the camera and maintains only the selected objects that are being represented, the rest are not retrieved from the database. The module is also in charge of performing the transformations in terms of scale, translation and rotation in order to obtain visual consistency in the scene. Finally, it captures the events made by the user on the screen through the GUI module of the user interface. The *View* also identifies different classes for enabling AR or VR. In the second, the camera attributes are the corresponding to the mobile device. The package *Graphics* is described in Section x.

Finally, the *Model* is in charge of the data management in the system and their topological relations. It attends to the information requests of the controller and sends them to the views. When the information is changed by the user though the *View*, these changes are received by this module and stored in the corresponding tables in the database. The set of geometric classes are embedded in the package *Core*, which will be described below.

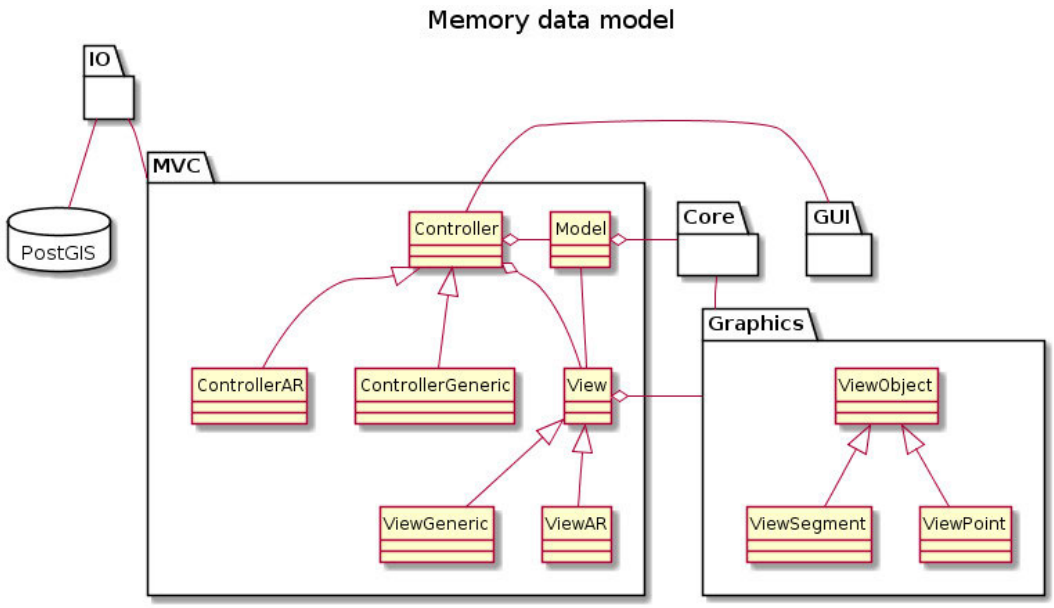


FIGURE 4. Packages and classes defined in memory.

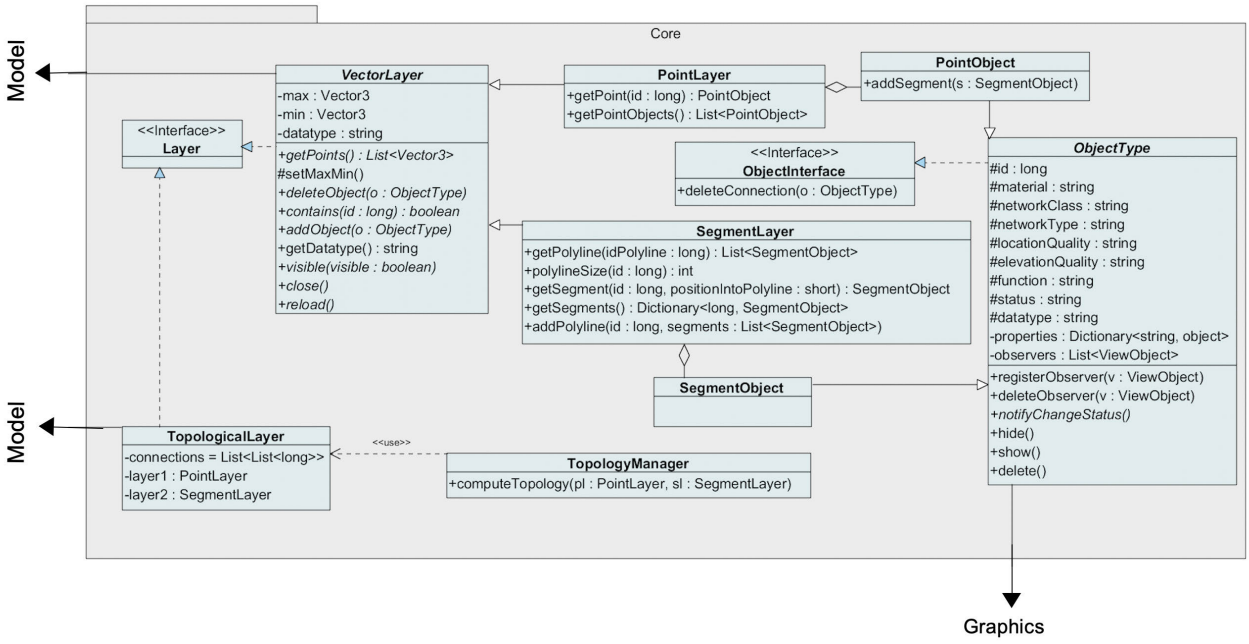


FIGURE 5. The core of geometrical classes in memory.

The topological structure of nodes and distribution elements must follow the same scheme of the database. There is only one important difference in order to achieve greater definiteness when dealing with any geometric primitive. This implies the access to edges and vertexes in order to be selected and edited individually, if necessary. The type used in the PostGIS database for the definition of polylines is *linestring*. Therefore in memory, this *linestring* is decomposed into the set of its vertexes in order to be geometrically edited independently while preserving the network topology. However, the definition of node-type objects described as 3D points on disk and memory does not change. Section V describes the CRUD operations according to this data model.

Figure 5 visualizes the clases of the *Core* package. It maintains a direct relation with the class *Model* and the package *Graphics*, as Figure 4 shows. All geometric elements in the data model are defined as *ObjectType*. This abstract class maintains the generic description of them, including the

TABLE 1. List of adjacency representing the graph of Figure 3.

ObjectType	list of adjacents
a	1
b	1,2,5,4
c	4
d	2,3
e	3
f	5
1	a,b
2	d,b
3	e,d
4	b,c
5	f,b

reference to the object visualized in the scene. An *ObjectType* object is instantiated as *PointObject* or as *SegmentObject*. In the first type we include wells or manholes in the sewage system, or interruptors in electrical networks. In the second, depending on the installation, there are pipes or cables. The *SegmentLayer* and the *PointLayer* classes represent collections of *SegmentObjects* and *PointObjects*. Each of these layers of geometric objects are defined as classes inherited from the class *VectorLayer*, which maintains the bounding box of the objects included and the type of data represented.

Topology is represented with the class *TopologicalLayer*. This class defines a graph of connected *ObjectType* elements as an adjacency list [45]. Every time an *ObjectType* object is selected in the *View* (referred as *GameObject* in the view), the *Model* identifies the corresponding geometric object. Then the graph identifies all the objects connected. This way when an element is selected for being translated, as described in Section V-B, all objects directly connected are known. As a matter of fact, all objects connected through others are also known by means of any graph path algorithm.

Table 1 represents the adjacency list of the example of Figure 3, in which node-type and distribution elements (point and segment types) are represented. Direct connected elements are defined in the same list, for instance the node-type element denoted as *b* are connected with the pipes {1,2,5,4}. The rest of connections, if required, can be located by using the DFS algorithm described above.

C. CityGML DATAMODEL

Once the data is stored in the spatial database and managed in memory directly by the user, the model should be able to be exported to standardized data models for information exchange and external viewing in any GML viewer.

Among the different standards for the representation of urban space, we choose CityGML because of its geometrical, topological, semantical and appearance properties for the most relevant topographic objects in cities and regional models. Even though the CityGML covers several classes, some others such as distribution facilities are not yet specifically modeled in the standard. However, the user can use the concept of ADE (Application Domain Extension) to extend the schema with new classes and attributes which are not explicitly modeled in CityGML.

1) SIMPLIFIED UTILITY NETWORK ADE FOR THE SEWAGE SYSTEM

The core of the Utility Network ADE is described in [44]. It defines the below-ground urban facilities by modeling the elements that compose it and their mutual relations explicit, whereas the entire network topology and topography is also represented. This core model has been extended by additional classes that also describe the entities of any utility network in a semantical-functional way [41]. In this core scheme, the abstract class *_NetworkFeature* inherits from the CityGML class *_CityObject* and represents the semantic or thematic description of any network entity. Therefore, a *Network* is a collection of these *_NetworkFeature* objects, although it might be structured into *subNetworks*. These two classes are used for the topographic representation of utility networks. In addition, each *_NetworkFeature* can consist of other objects of the same class as a way of modeling component hierarchies.

The representation of the network topology and connectivity is performed by dual classes: *FeatureGraph* and *NetworkGraph*. Therefore, the *FeatureGraph* represents each utility in a separate graph structure representing its functional, structural and topological aspects. A *FeatureGraph* object is in fact a *Node* or an *_Edge*, a dual representation for point-type and segment-type elements respectively. The edges in this data model can be classified into *InteriorFeatureLink* and *InterFeatureLink* to differentiate between the same *NetworkFeature* or to establish the connectivity to other *NetworkFeature* elements. In the same way, different instances of the class *Network* can be linked by *NetworkLink* objects. The geometric representation of nodes and edges are the GML data types *gml::Point* and *gml::_Curve*.

This core data model and its extension to any below-ground facility has numerous advantages in terms of completeness, standardization and covers the goal of providing the topological, topographic and functional representation of these infrastructures. The major drawback we have found is the little impact it has had on a practical level. Despite all the advantages mentioned above, it has not been widely used in real projects, and has had a relative impact in the academic field. One of the reasons may be the fact that said data model must also exist in a spatial database and in the memory of a computer, as we propose in this article. The CityGML model is interesting for visualization and as exchange format. For real-time interaction with the model and in order to carry out the CRUD operations mentioned above, a conversion mechanism must be established.

In this paper, we propose a simplified data model for the Utility Network ADE and the subclasses modeled as specializations of the *NetworkFeature* described in [41]. This data model reduction is due to the explicit adaptation to our input data, in which sub-nets are not presented. Some attributes used in our data base and memory model have also been added to make them compatible. Figure 6 shows this simplified CityGML data model.

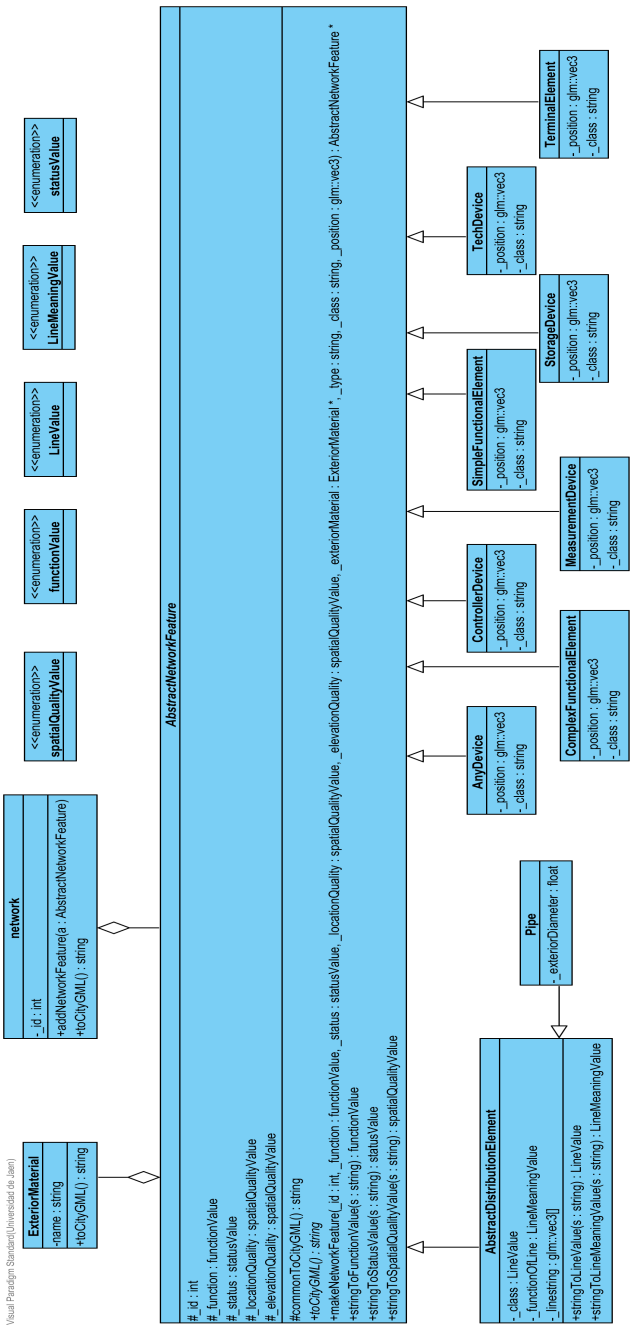


FIGURE 6. Class diagram for the simplified CityGML Network ADE extension.

As in the initial proposal, different subclasses inherit from *_NetworkFeature*, the class that represents any infrastructure element, either node or pipe type. An *AbstractDistributionElement* is also a child class which represents any type of conduit. For simplification reasons, the class *Pipe* is defined with its exterior diameter, following the same scheme described in the database datamodel of Figure 2. Wires, tunnels or any type of distribution element should also inherit from this abstract class.

Node-type elements may be diverse, as well as in [41]. Following the Utility Network ADE, they can be cataloged into

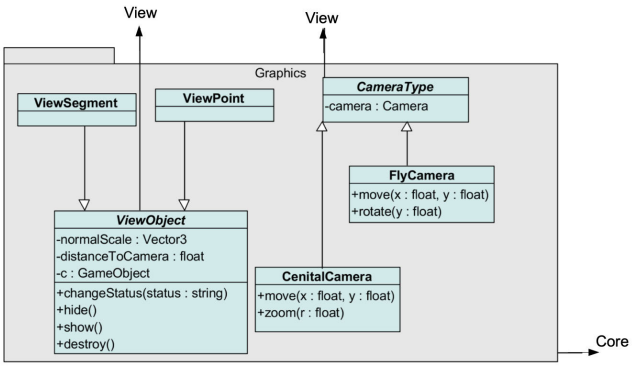


FIGURE 7. Classes associated to the graphics package.

different classes such as: *SimpleFunctionalElement*, *MeasurementDevice*, *TerminalElement*, *ControllerDevice*, etc. These classes provide additional information at functional level. Then, a well can be considered a *SimpleFunctionalElement*, a valve is a *ControllerDevice* and a corrosion detector is a *MeasurementDevice*. Some additional enumeration classes define, for instance, the current status of the infrastructure element (*statusValue*) or the accuracy of the value assigned (*spatialQualityValue*), in case this value has been re-edited after a visual inspection. Some other possibilities are to be interpolated, etc.

In summary, what our approach has in common with those defined in [41], [44] is the general structure and the terminology used. In fact, the main idea is to make this proposal compatible with the one defined by Becker et. al., by adding complexity and new types of facilities as they are needed. In contrast, we add additional topological information from node-type elements towards the pipes that are annexed.

V. FACILITIES FUNCTIONALITY THROUGH DIGITAL REALITY

Once the system is fed with data from the sewer network, then navigation and CRUD operations are available. The way to access and manage the information in this system is through its three-dimensional graphic representation on the screen. Then, the user is able to navigate and interact with the graphical elements in the scene. Additional operations such as network connectivity are also available through the GUI. The data model designed in memory allows these operations to be carried out in real time.

A. VISUALIZATION AND NAVIGATION

The diagram of Figure 4 depicts different modules for visualizing the scene. Both the *Controller* and the *View* handle the environment differently depending on whether the scene is represented with virtual reality (Generic) as if it is carried out with augmented reality (AR).

1) VIRTUAL REALITY

Regarding virtual reality, visualization is performed in OpenGL in the *Graphics* module of classes by means of

Shaders. This programmable pipeline is advantageous for greater performance gain. Each infrastructure element stored in the data models is placed in the scene by means of specific 3D models representing pipes or any other facility element. Their exact location stored initially in the database is replaced by a transformation matrix which provides the translation, rotation and scale of each of them.

Visualization efficiency is also increased with the use of the pattern *Flyweight*. A flyweight object type minimizes memory usage by sharing as much data as possible with other similar objects. In this way, the representation of all objects with the same aspect is stored in memory just one, but replicated through the scene for each feature element. Therefore, the geometric data is transformed into 3D models, a different one for each type of underground infrastructure item to be displayed. In this manner, there is a single model for all pipes, each of them is translated, rotated or scaled according to this transformation matrix.

Navigation is in fact implemented through camera movements. There can be several camera types depending on the way of navigation. In particular, we implement the *OrbitCamera* and the *WalkCamera*. The first allows free turns of the camera and is designed to have a free navigation and a global view on the scene. The second restricts the rotation, so that it only allows top-down movements and navigation on the XZ plane once the height has been chosen. This facilitates the inspection of infrastructures at a given height level.

Some examples of virtual reality are depicted in Figure 8 in which the sewage system and the relief of the terrain are represented. The infrastructure can be inspected visually, the scene can be easily navigated or edited as described below. The most common devices for this type of interaction are desktop devices.

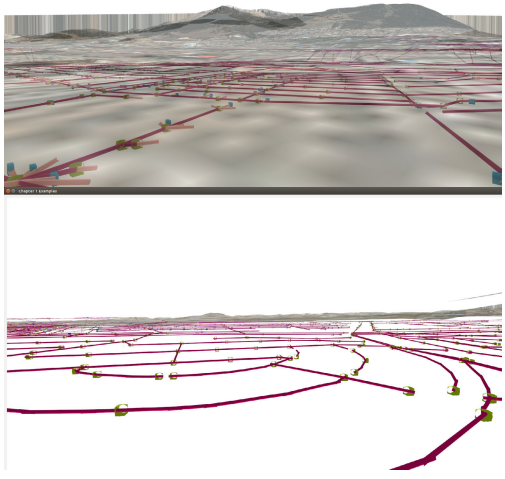


FIGURE 8. Virtual reality for underground facilities.

However, visualization and interaction can be performed using WebGL (Web Graphics Library), a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser. Figure 9 shows an example of WebGL

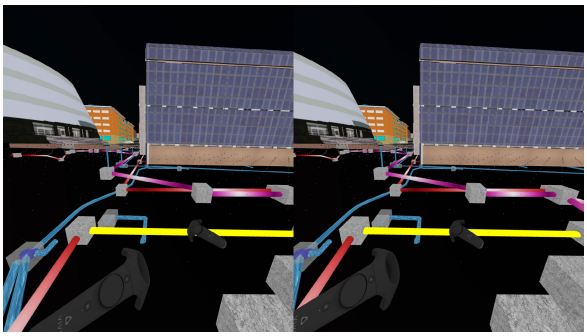


FIGURE 9. VR view for the management of the underground infrastructure.

scene of underground infrastructure. There are two separate images for each eye, providing 3D sensation thanks to the stereoscopic graphics rendering when displayed on the phone with the headset.

2) AUGMENTED REALITY WITH ARCore

The major advantage of using the separated design in Figure 4 for visualization is that it can be changed to be adapted to the specific client device. For instance, it can be re-designed to enable augmented reality or mixed augmented reality. In this case, the client devices are mobile devices working on site, allowing to virtually eliminate the asphalt or wall layers to visualize the infrastructures. This has many advantages such as knowing exactly the location of these elements.

In this case, the visualization module is implemented using Unity and ARCore. Unity is a cross-platform real-time video-game engine which has gained immense popularity. ArCore is the Google′s platform for providing augmented reality experiences. This technology allows the phone to sense, understand and interact with the information about the environment. ARCore uses some capabilities of advanced phones such as motion tracking to detect its position and orientation as it moves through space. In our solution, this is interesting to virtually visualize the hidden infrastructures and interact with them through the clicking them on the phone's screen.

Figure 10 shows AR enabled on the sewerage system. In the first image a well and the pipes associated are displayed on the correct position. On the street, operators can learn the exact position and orientation of the pipes in case they need to dig into the ground surface. The innovation that Mixed Augmented Reality (MAR) raises in this context through the matching between real and virtual world. This implies the geometric alignment between two spaces in order to define their correct scale and size. Outdoors, the Global Positioning System (GPS) signal is taken as the reference for the initial position adding measurements of a sub-metric GPS. In order to correct orientation errors, an assisted method allows to rotate the virtual world until this matching is finally obtained, as described in [30]. Once this matching has been obtained, the phone motion tracking sensors are able to maintain this concordance.

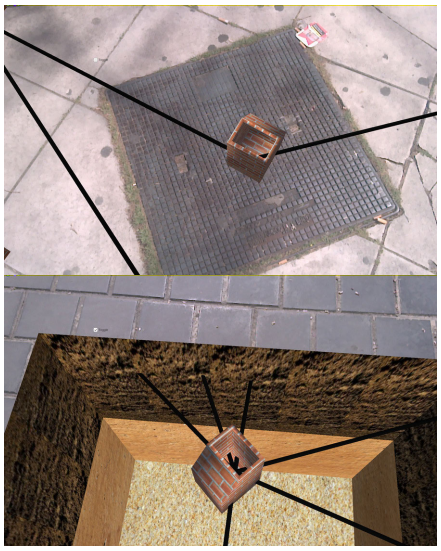
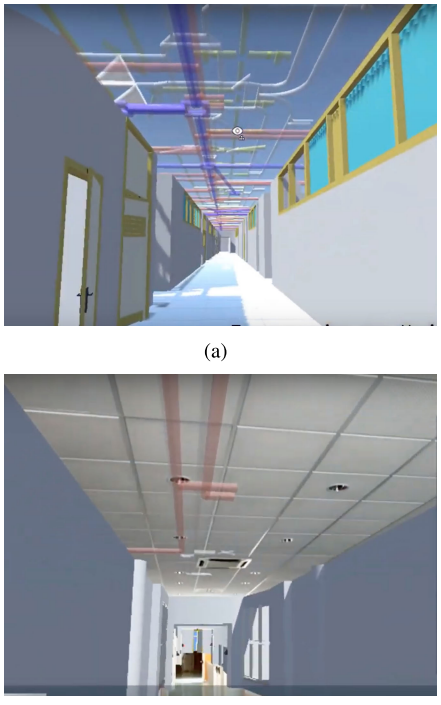


FIGURE 10. Augmented reality for underground facilities.



(a) (b)

FIGURE 11. Mixed reality enabled indoors.

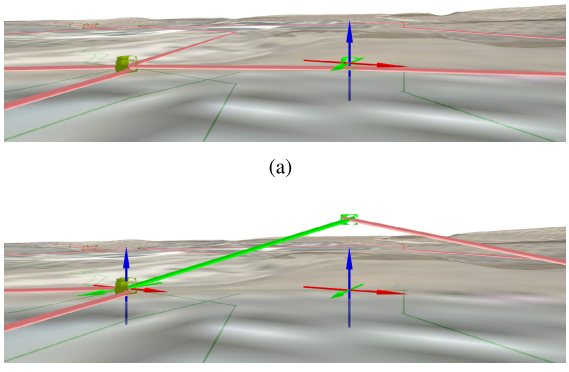
Similarly, it is also possible to enable mixed reality indoors. Again, this means to put together virtual and augmented reality, allowing the user to match real and virtual architecture. Layers of walls and facilities can be enabled while walking around the building. Figure 11(a) shows a virtual representation of walls and indoor facilities. In this case, pipes and wires are embedded in the ceiling. There are layers of lighting, ventilation, communications networks and air conditioning. Walls are also displayed in Figure 11(b), but the actual ceiling is also visible in one of the layers. Each of these layers of virtual elements can be enabled or disabled to provide depending on the functionality desired.

B. GEOMETRY SELECTION AND EDITING

During the navigation process, the user is able to select any infrastructure element by a simple click on the mouse or screen. Selection is normally the first step to perform the CRUD operations described above, that is, creation, deletion or editing. Again, the data models described in this paper facilitate these types of tasks thanks to their topological design.

The treatment of data for CRUD operations performed by filling in the records is also available in our prototype system but not discussed in this article, since they are the classic access method. We focus on the interaction process performed directly through the scene in DR, either enabling virtual or augmented reality. As stated above, this selection process is mainly done by clicking on the screen using a mouse or directly on touch screens. This click implies the selection of the first visible geometry located in the touched position. This method is performed in memory following the methodology described in [46] for ray-traversal algorithms. An *octree* of the whole scene is computed and stored in memory in order to accelerate the process. The click generates a ray from the click position towards the direction where the camera faces the interior of the scene. This ray may collide with the geometry inside the scene. If this intersection finally occurs, the collided geometry is returned to the user.

After selecting the geometric representation of the pipe or node-type element, deletion or editing operations are available. Even the creation of new elements is done by selecting other infrastructures to which they are connected.



(a) (b)

FIGURE 12. Geometry editing into a VR interface.

1) VIRTUAL REALITY

Regarding editing in environments with virtual reality, the process is helped by a 3D axis depicted in Figure 12. Only vertexes are allowed to be edited in one of the

three dimensions at a time by selecting one of the orthogonal axes. Then any of these axes can be dragged in order to move the element in one of the selected directions. Figures 12(a) and 12(b), respectively, show an example before and after this editing process. In addition, the two portions of pipes connected are also selected and moved to accompany that movement. Additional orthogonal axes in annexed node-type elements are added to continue editing, if necessary. In the case of moving a well or any other node-type element, their associated pipes are also dragged around while detecting the intersection with the rest of elements around. This guarantees the connection while the topographic restrictions are considered. All this functionality is also important in the design phase, in order to place accurately the new facilities considering the terrain depth and its interrelation with the rest of subsoil elements.

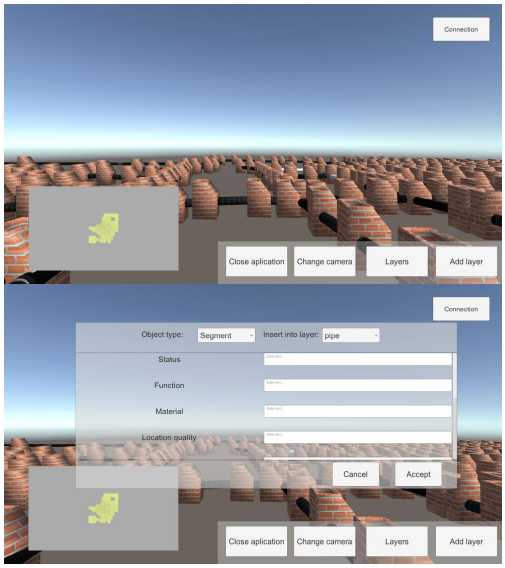


FIGURE 13. Examples of the interface during some CRUD operations.

Figures 13 and 14 show some instants during the navigation process and the CRUD operations though the application interface. There are different levels of detail (LOD) automatically displayed according to the type of camera and its position. In Figure 13, the scene is navigated through the GUI controls (mouse and arrow keys). At any time, a click on any infrastructure element can be done so that it is selected. Then, any of the CRUD operations can be called. In this case a new section of the pipe is added when an attached pipe that already exists is selected. The rest of the process consists of filling a form with another series of parameters of non-geometric nature, as visualized in the same figure.

In Figure 14, a section of the pipe is selected, and both extreme points highlighted. Then, a menu overprinted on the screen provides the full information stored in the database. This also provides the option of editing or deleting the selected element, as the right-most figure shows.

FIGURE 14. Zenith view for CRUD operations.

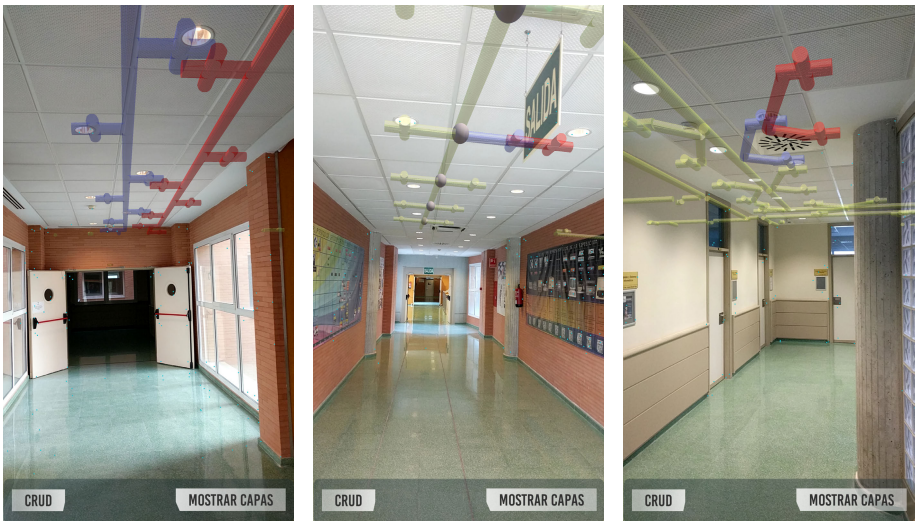
2) AUGMENTED REALITY

Selection is also the first step for deleting or editing geometry in a mobile device with augmented reality. In this case, it is made with the gesture *Tap*, consisting of a quick touch in the screen with the finger. It is equivalent to the left mouse click. Then, the specific layer of elements or a simple portion of them is selected, depending on the application. In Figure 15(a) an entire portion of facility is selected, coloring the selected set of facilities in deep red. For that purpose, the graph is traversed to find all annexed elements under a specific condition. Then, the gesture *one-finger-drag* allows to move the set by making a copy of the original in blue. After that, the user can decide whether to make the changes effective or not. In Figure 15(b), the selection is performed only in a small section of the pipe. Then, it is moved in order to connect it with the rest of network. This way, corrections to network connectivity can be performed in situ, as stated in following section. In this case topology is used to determine the elements directly connected to the selected pipe.

C. NETWORK CONNECTIVITY

Connectivity check is one of the direct applications of a topological data model in network facilities, as discussed in Section IV-A. For instance, the sewer network is known to be fully connected, so that all the water goes to a single and large collector. However, after checking network connectivity, a graph with several components is obtained. On the other hand, there can be different electrical networks in a building that provide different power level. Even so, each of these networks must be properly connected at the individual level.

These errors may be due to the lack of geometry or due to the precision with which the topology is generated,



(a) (b) (c)

FIGURE 15. Selection and movement of pipes with augmented reality.

FIGURE 16. Checking and repairing connectivity.

as described in Section I. The algorithm also works by clicking on any portion of the infrastructure, in which case the non-connected components are depicted with a different color. The example of Figure 16 shows a portion of this sewer network. Those non-connected portions to the main are depicted in dark red. After analyzing the cases marked with circles in the bottom figure, we can observe that case *A* has two different anomalies. The first is due to lack of geometry in the input data. In the referred position there should be a node-type element. The second is due to some inaccuracy when finding the neighborhood connection because of having input values greater than the Epsilon used to join the geometry. The solution is carried out by creating and editing the geometry according to the guidelines given by the utility providers. After this editing process, the resulting network is displayed completely green after finding real connectivity. Case *B* is also compared with reality. The problem detected is the lack of a node-type object in the intersection point. The following step is to determine the exact element and insert it in order to make the network cohesive.

The process is similar to augmented reality. Figure 15(c) shows a portion of a disconnected component of conduits in red and the final location after moving to the correct position. Once the process is performed, the network is a single connected component.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present three data models for representing and managing hidden facilities. The design maintains the topology of a network represented as a graph, as well as its topography. In fact, it is the same data model but adapted to be represented in a spatial database, in the memory of a computer or mobile device, and also as an interchange format in CityGML.

The spatial database is implemented in PostGIS, which provides spatial objects for the PostgreSQL database. This database model provides 2D and 3D data types and operations on that spatial data and also supports all the objects and functions specified in the OGC. The usual access process is through SQL queries from any device, for which it is considered a client-server architecture with the database hosted

on the server. This information stored in the database must be able to be visualized, consulted and edited ubiquitously in real-time and in a graphic environment, in order to take advantage of the potential of 3D data. To achieve this purpose, a data model in memory is also designed to be installed in the client device. This model follows the MVC pattern in order to facilitate and speed up the CRUD operations, such as editing or deleting infrastructure elements individually. This design allows the use of different technologies in the client, such as virtual or mixed augmented reality. Finally, the system must be able to interchange the information to the CityGML format. Thus, a new data model has been designed as an extension of CityGML by simplifying the UtilityNetworkADE.

These data models have been fed with real data from different facilities such as the sewerage system of a city or the whole set of conduits of a building. The topological design makes it possible to operate on the elements of this network for CRUD operations. Some additional tests have been done on this network at the level of connectivity.

The creation of the functional elements of different infrastructures should be considered for future research. This process does not differ from that corresponding to the sanitation network at the design level. However, at the level of analysis, it requires the participation of experts in the field. For example, the drinking water supply network has cut-off valves, measuring elements and pressure control that contribute to the physical behavior of the system. In order to achieve such functionality, the incorporation of the IoT technologies is facilitated thanks to the client-server architecture. In this way, the system transcends the information model and becomes a fully functional model on a physical level.

REFERENCES

[1] National, Academy, and Sciences, *Underground Engineering for Sustainable Urban Development*. Washington, DC, USA: The National Academies Press, 2013.

[2] M. Nagappan and E. Shihab, "Future trends in software engineering research for mobile apps," in *Proc. IEEE 23rd Int. Conf. Softw. Anal., Evol., Reengineering (SANER)*, vol. 5, Mar. 2016, pp. 21–32.

[3] F. Booty, Ed., *Facilities Manage*, 4th ed. Oxford, U.K.: Butterworth-Heinemann, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780750689779000190

[4] J. Curiel-Esparza and J. Canto-Perello, "Selecting utilities placement techniques in urban underground engineering," *Arch. Civil Mech. Eng.*, vol. 13, no. 2, pp. 276–285, Jun. 2013.

[5] S. Heragu, *Facilities Design* (Facilities Design). Boca Raton, FL, USA: CRC Press, 2016. [Online]. Available: https://books.google.es/books?id=9jF9PwAACAAJ

[6] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Hoboken, NJ, USA: Wiley, 2011.

[7] D. Ilter and E. Ergen, "BIM for building refurbishment and maintenance: Current status and research directions," *Struct. Surv.*, vol. 33, no. 3, pp. 228–256, Jul. 2015.

[8] P. Teicholz, Ed., *BIM for Facility Managers*. Hoboken, NJ, USA: Wiley, 2013.

[9] Y. Shi, J. Du, S. Lavy, and D. Zhao, "A multiuser shared virtual environment for facility management," *Procedia Eng.*, vol. 145, pp. 120–127, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705816300339

[10] S. Kim and J. Heo, "Development of 3D underground cadastral data model in korea: Based on land administration domain model," *Land Use Policy*, vol. 60, pp. 123–138, Jan. 2017.

[11] X. Li, Z. Lv, J. Hu, B. Zhang, L. Shi, and S. Feng, "XEarth: A 3D GIS platform for managing massive city information," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Jun. 2015, pp. 1–6.

[12] A. Abdul-Rahman, S. Zlatanova, and V. Coors, Eds., *Innovations in 3D Geo Information Systems* (Lecture Notes in Geoinformation and Cartography). Berlin, Germany: Springer, 2006. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-36998-1

[13] T. W. Kang and C. H. Hong, "A study on software architecture for effective BIM/GIS-based facility management data integration," *Autom. Construction*, vol. 54, pp. 25–38, Jun. 2015.

[14] C. Mignard and C. Nicolle, "Merging BIM and GIS using ontologies application to urban facility management in ACTIVe3D," *Comput. Ind.*, vol. 65, no. 9, pp. 1276–1290, Dec. 2014.

[15] H. Masuda, "Topological operators and Boolean operations for complex-based nonmanifold geometric models," *Computer-Aided Design*, vol. 25, no. 2, pp. 119–129, Feb. 1993.

[16] P. van Oosterom and S. Zlatanova, *Creating Spatial Information Infrastructures: Towards the Spatial Semantic Web*, 1st ed. Boca Raton, FL, USA: CRC Press, 2008.

[17] D. Xu and S. Zlatanova, "An approach to develop 3D Geo-DBMS topological operators by re-using existing 2D operators," in *Proc. ISPRS*, Nov. 2013, pp. 291–298. [Online]. Available: http://resolver.tudelft.nl/uuid:3288ef8b-b85c-46c5-b77c-03779a20ba6d#.VegCWrTyLuE.mendeley

[18] C. T. Khuan, A. Abdul-Rahman, and S. Zlatanova, "New 3D data type and topological operations for Geo-DBMS," in *Proc. Urban Regional Data Manage. UDMS Annu.*, 2008, pp. 211–222.

[19] Y. Hu, S. Ravada, R. Anderson, and B. Bamba, "Topological relationship query processing for complex regions in oracle spatial," in *Proc. 20th Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*, 2012, pp. 3–12.

[20] R. O. Obe, *PostGIS in Action*, 2nd ed. New York, NY, USA: Manning, 2015.

[21] S. Daum and A. Borrmann, "Processing of topological BIM queries using boundary representation based methods," *Adv. Eng. Informat.*, vol. 28, no. 4, pp. 272–286, Oct. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474034614000391

[22] M. Liotine, *Mission-Critical Network Planning* (Artech House Telecommunications Library). Boston, MA, USA: Artech House, 2003.

[23] L. Yuan, Z. Yu, W. Luo, L. Yi, and G. Lü, "Multidimensional-unified topological relations computation: A hierarchical geometric algebra-based approach," *Int. J. Geographical Inf. Sci.*, vol. 28, no. 12, pp. 2435–2455, Dec. 2014. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/13658816.2014.929136#.VbitMU3tlBc

[24] M. S. Khan, H.-Y. Kang, J.-H. Lee, D.-H. Park, and D. Kim, "Semantic Data Representation Based on Risk Index for Monitoring Underground Facility," *Int. Inf. Inst.*, vol. 20, no. 2B, p. 1269, 2017.

[25] J. M. Torres, L. Duenas-Osorio, Q. Li, and A. Yazdani, "Exploring topological effects on water distribution system performance using graph theory and statistical models," *J. Water Resour. Planning Manage.*, vol. 143, no. 1, Jan. 2017, Art. no. 04016068.

[26] J. Underwood, U. Isikdag, J. Underwood, and U. Isikdag, *Handbook of Research on Building Information Modeling and Construction Informatics: Concepts and Technologies*, 1st ed. Hershey, PA, USA: IGI Global, 2009.

[27] K. Ying, J. Chu, J. Qu, and Y. Luo, "A model and topological analysis procedures for a pipeline network of variable connectivity," *Adv. Eng. Softw.*, vol. 48, pp. 40–51, Jun. 2012, doi: 10.1016/j.advengsoft.2012.02.013.

[28] M. Wu, B. Guimei, Y. Wen, J. Shen, and T. Chen, "A 3D topology model for multiple topological representations based on combinatorial maps," in *Proc. 18th Int. Conf. Geoinformatics*, Jun. 2010, pp. 1–4. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-77958060538&partnerID=tZOtx3y1

[29] S. Meža, Ž. Turk, and M. Dolenc, "Component based engineering of a mobile BIM-based augmented reality system," *Autom. Construction*, vol. 42, pp. 1–12, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580514000363

[30] G. Soria, L. M. Ortega Alvarado, and F. R. Feito, "Augmented and virtual reality for underground facilities management," *J. Comput. Inf. Sci. Eng.*, vol. 18, no. 4, Dec. 2018, Art. no. 041008.

[31] X. Zhang, Y. Han, D. Hao, and Z. Lv, "ARGIS-based outdoor underground pipeline information system," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 779–790, Oct. 2016.
[32] G. Schall, S. Zollmann, and G. Reitmayr, "Smart vidente: Advances in mobile augmented reality for interactive visualization of underground infrastructure," *Pers. Ubiquitous Comput.*, vol. 17, no. 7, pp. 1533–1549, Oct. 2013.
[33] S. Lee, J. Suh, and H.-D. Park, "BoreholeAR: A mobile tablet application for effective borehole database visualization using an augmented reality technology," *Comput. Geosci.*, vol. 76, pp. 41–49, Mar. 2015.
[34] E. Stylianidis, E. Valaria, K. Smagas, A. Pagani, J. Henriques, A. Garca, E. Jimeno, I. Carrillo, P. Patias, C. Georgiadis, A. Kounoudes, and K. Michail, "LBS augmented reality assistive system for utilities infrastructure management through galileo and egnos," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLI-B1, pp. 1179–1185, Jun. 2016.
[35] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile augmented reality survey: From where we are to where we go," *IEEE Access*, vol. 5, pp. 6917–6950, 2017.
[36] G. Soria, L. M. Ortega, and F. R. Feito, "Google Tango outdoors. Augmented reality for underground infrastructures," in *Proc. 27th Spanish Comput. Graph. Conf. (CEIG))*, Seville, Spain, Jun. 2017, pp. 31–40.
[37] G. Schall, S. Zollmann, and G. Reitmayr, "Smart vidente: Advances in mobile augmented reality for interactive visualization of underground infrastructure," *Pers. Ubiquitous Comput.*, vol. 17, no. 7, pp. 1533–1549, Oct. 2013.
[38] T. H. Kolbe, G. Gröger, and L. Plümer, "CityGML–Interoperable access to 3D city models," in *Geo-information for Disaster Management*. Berlin, Germany: Springer-Verlag, 2005, pp. 883–899.
[39] T. H. Kolbe, "Representing and exchanging 3D city models with CityGML," in *3D Geo-Information Sciences*. Berlin, Germany: Springer-Verlag, 2009, pp. 15–31.
[40] T. Becker, C. Nagel, and T. H. Kolbe. (2010). *CityGML Extension for Modelling Utility Networks in 3D City Models*. [Online]. Available: http://www.citygmlwiki.org/index.php?title=UtilityNetworkADE_Version_0.1.0
[41] T. Becker, C. Nagel, and T. H. Kolbe, "Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization," in *Progress and New Trends in 3D Geoinformation Sciences*, J. Pouliot, S. Daniel, F. Hubert, and A. Zamyadi, Eds. Berlin, Germany: Springer, 2013, pp. 41–62. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29793-9_3
[42] A. Krüger and T. H. Kolbe, "Building analysis for urban energy planning using key indicators on virtual 3D city models—The energy atlas of Berlin," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 39, no. B2, pp. 145–150, 2012.
[43] I. Hijazi, T. Kutzner, and T. H. Kolbe, "Use cases and their requirements on the semantic modeling of 3D supply and disposal networks," in *Proc. Kulturelles Erbe erfassen und bewahren-Von der Dokumentation zum virtuellen Rundgang Wissenschaftlich-Technische Jahrestagung de (DGPF)*, 2017, pp. 288–301.
[44] T. Becker, C. Nagel, and T. H. Kolbe, "Integrated 3D modeling of multi-utility networks and their interdependencies for critical infrastructure analysis," in *Advances in 3D Geo-Information Sciences*. Berlin, Germany: Springer, 2011, pp. 1–20.
[45] M. T. Goodrich and R. Tamassia, *Algorithm Design and Applications*. Hoboken, NJ, USA: Wiley, 2015.
[46] S. F. Frisken and R. N. Perry, "Simple and efficient traversal methods for quadtrees and octrees," *J. Graph. Tools*, vol. 7, no. 3, pp. 1–11, 2002.

LIDIA M. ORTEGA received the B.Sc. degree in computer science from the University of Granada, Spain, and the Ph.D. degree from the University of Seville. She has been an Associate Professor with the Department of Computer Science, University of Jaén, teaching at the High Polytechnics Institute, since 1990. Her research interests include computational geometry applied to computer graphic, 3D GIS, spatial databases, and geomatics.

JUAN M. JURADO is currently a Lecturer with the University of Jaén. He belongs to the Research Group of Computer Graphics and Geomatics (TIC-144), University of Jaén. His line of research is framed in the area of computer graphics, more specifically, in digital image processing, spectral response of real biological systems, fusion of heterogeneous data, and study of predictive models based on deep learning. This requires the creation and management of space-time databases, as well as the application of I. A. like neural networks.

JOSÉ LUIS LÓPEZ RUIZ was born in Linares, Spain, in 1993. He received the degree in computer science from the University of Jaén, in 2018, where he is currently pursuing the Ph.D. degree with the Research Group of Computer Graphics and Geomatics (TIC-144). His research interests include computer graphics, 3D GIS, and augmented reality.

FRANCISCO R. FEITO received the B.Sc. degree in mathematics from the University Complutense of Madrid and the Ph.D. degree in computer science from the University of Granada, Spain. He was the Head of the Department of Computer Science, University of Jaén, Spain, from 1993 to 1997 and 2003 to 2008, a Vice Chancellor in charge of studies and quality, from 1997 to 1999, and a Vice Chancellor in charge of research and international affairs with the University of Jaén, from 1999 to 2002. He is currently a Full Professor with the Department of Computer Science, University of Jaén. He has been teaching with the High Polytechnics Institute, Jaén, and the Head of the Department and the Graphics and Geomatics Research Group. His research interests include formal methods for computer graphics, geometric modeling, computational geometry, geographical information sciences, virtual archaeology, and precision agriculture.