MANAGEMENT AND 3D VISUALIZATION OF PIPELINE NETWORKS USING DBMS AND AEC SOFTWARE

Yunqiang Du[1], Sisi Zlatanova[2] and Xianglin Liu[3]

[1] The urban planning bureau of Yibing city, Sichuan Province, P.R.China
yqdu2003@yahoo.com.cn

[2]GISt, OTB Research Institute for Housing, Urban and Mobility Studies
Delft University of Technology, Jaffalaan 9, 2628 BX Delft, The Netherlands
s.zlatanova@tudelft.nl

[3]DEOS, Faculty of Aerospace Engineering, Delft University of Technology
Kluyverweg 1, 2629 HS Delft, The Netherlands
x.l.liu@tudelft.nl

Commission IV/8

KEY WORDS: Databases, CAD, GIS, Data structures, Modelling, Visualisation

ABSTRACT:

Management and visualization of underground utilities have been always of a great concern in many countries. Insufficient, inaccurate and unclear information about location and depth of cables and pipelines may cause various problems and may even result in tragic accidents. In this paper we argue that 3D visualization of pipelines is of a critical importance for efficient maintenance (in the office and on the field), providing a better perception and understanding of the complexity of the underground networks. Indeed, management of pipes as 3D objects (e.g. cylinders) is rather difficult and unrealistic. This paper presents our approach for 3D visualization of pipelines organized as 3D lines in a DBMS. Parameters, such as diameter, heights, radius, etc. are organized together with the pipe geometry as well. The 3D effect is obtained on the fly prior the visualization at the front-end. The 3D shape of the pipelines is computed from the centrelines and the additionally stored parameters. Various tests are performed on a case study area of Yibing city, Sichuan Province, P.R. China and experimental data set of the Netherlands.

1   UTILITY MANAGEMENT IN PROGRESS

Pipelines such as water supply, sewage, power supply, heat supply, industrial pipelines and communication lines, are essential infrastructures in cities. They provide substantial material basis for cities by transporting water, energy and information. Being an important part of daily rhythm of a city, management of utility networks has been always challenging due to numerous issues.

Often insufficient documents are available describing previously existing pipelines or some of those documents are not accurate enough. This tendency has been increased in the last decade while rebuilding and substituting of existing principal pipelines to fit in with increased demands of citizens and industrial development. Moreover, many new designed underground pipelines for different purposes have been built along with many other engineering projects.

Many local governments are still not able to control completely this economical booming. Responsible construction companies of engineering projects might hardly make a full investigation to obtain specific documents from management agencies related to existing pipelines before they start engineering projects. Weak-training workers are regularly reported for their wrong operations during construction. Many serious accidents such as the cut-off of gas, water and heat supplies, communication lines and even the overflow of sewage were caused by blind-cutting or fault damaging. For example, in China it has been estimated

an economic loss of up to two hundred million US dollars per year during eighties and the beginning of nineties (Ministry of construction in China, 2005). Similar investigations are completed in other countries (e.g. Roberts et al 2002, Penninga and Oosterom 2006) and the results are striking.

The significant economic loss and inconvenience caused by inefficient management of pipelines has attracted an unprecedented attention in communities of governmental and technological agencies. In China, the first successful and quite comprehensive information system of urban underground pipelines was set up for Xiamen city in Fujian province in 1995. This system was developed using Arc/Info UNIX platform. It was not only able to carry out some GIS fundamental functions, such as check-in of raw data, graphics generating, mapping browsing, inquiring, editing, map output and database management, but also some specific functions for pipelines, for example, analysis of vertical and horizontal profiles, spatial analysis of intersection and analysis of broken pipeline. Since then, many other cities in China have developed or upgraded their systems to similar Underground Pipelines Information Systems (UPIS, a GIS for specific purpose of pipelines management) by using different kinds of GIS Desktop platforms, like Arcview, Mapinfo, Mapgis and GeoStar (two of China's own GIS platforms). This situation has further developed into that almost all the systems were updated by network UPIS for meeting the need of simultaneous operations from multi-operators, multi-users and multi-agencies. Some of cities even applied web GIS technology in their pipelines

management systems by using Client/Server and Browser/Server network infrastructure.

However, from a visualization point of view all systems need improvement. All pipelines data are generally projected into certain a horizontal plane (2D) though they might be displayed in format of vertical profile (2D) at some limited key places. This is certainly not sufficient, particularly in the case of many pipelines intersecting each another vertically and horizontally. For horizontal 2D maps generated by current UPISs, those pipelines with the same *x,y* coordinates, but different heights, are rather difficult to understand even by professionals, let alone many other weak-training workers in China who carry out most specific field work.

This paper presents our approach for 3D visualization of pipelines. The pipes and cables are maintained with their centrelines in DBMS and visualized in Architecture, Engineering and Construction (AEC) software for inspection (in 3D) and editing (in 2D). The implementation is realized for Oracle Spatial 10g and Microstation V8. Section 2 discusses the possibilities for utility data organization in Oracle Spatial. Section 3 presents the selected data structure. Section 4 elaborates on the visualization approach considering cylindrical and rectangular pipelines and construction of special connections at 3D symbols. Last section summarizes the results of the tests and addresses further research.

2 GEOMETRIC MODELS OF ORACLE SPATIAL 10G APPLICABLE FOR PIPELINES

Spatial objects can be currently managed in Oracle Spatial using four different models namely *geometry*, *topology*, *Linear Referencing System (LRS)* and *network* (Oracle 2003). Besides the data types, an extended set of spatial function and operations is attributed to each data model. The models are not equally appropriate for modelling utility networks. The topology model has *node*, *edge* and *face* data types organised in a structure similar to the well-know *wheel-chain* data structure (Penninga, 2004). The major disadvantage of the topology model is that it needs area objects (i.e. faces), which are commonly not available in utility networks. The network model and LRS are typical graph data structures and conceptually very appropriate for maintenance of utility networks. However, there are few practical drawbacks. For example, LRS is not supported by any front-end application. To be able to visualize spatial objects managed as LRS data types they have to be converted to the data types of the geometry model (using Oracle Spatial function). The network model avoids this obstacle by maintaining the geometry within network data types (i.e. *node*, *link* and *patch*) and, therefore, it is readily accessible for visualization. Though very attractive, the model requires development of user-defined scripts for populating the tables and checking the consistency, which was outside the scope of this research. Therefore we have concentrated on the geometry model. The geometry model (i.e. SDO_GEOMETRY) has several simple geometry types (and collections of them) such as *point, point cluster, line string, polygon, arc line string, arc polygon, compound line string, compound polygon, circle, optimized rectangle*, and even optionally, it can be indicated that a certain geometry is *unknown*.

Pipelines and cables mainly include cylindrical and rectangular pipelines, e.g. cylinder and ditch with cover. Currently, such geometric types are not available in Oracle Spatial and therefore 3D pipelines have to be stored either as: 1) combinations of primitive geometry types, or 2) user-defined types. The common disadvantage of user-defined types is impossibility to use the spatial functionalities of DBMS such as spatial indexing

and spatial operations. This motivated us to use only existing data types.

One option to maintain cylinders and rectangular pipes with existing data types would be to simulate them as reported in Yong 2003. The basic idea is that a cylinder can be defined as a group of flat quadrangles connected each other. The quadrangles are in 3D and 3D vertexes represent them. Based on this idea, a rectangle pipelines can be exactly expressed with four polygons, but a cylindrical pipeline will be described with much more than four polygons. Indeed, it is quite unrealistic and inefficient to store all these polygons into Oracle Spatial.

In this paper we follow an approach, which considers the mathematical definition of cylinder, i.e. the cylinders spatial shape is only described by a centreline and a diameter. This allows us to consider that a cylinder can be created in Microstation using these two parameters. As for rectangle pipeline, it can be described with a centreline and its size considering height and width of a ditch. In this way, pipelines can be sorted in Oracle Spatial using the supported 3D line data type.

The advantages of this approach are:

- Easy to manage. Line is a simple geometric type, its input, edit and update are easy.

- Easy to implement. Create, modify, validate and query of lines are a trivial operation. Moreover spatial functions such as length, overlap, intersect, etc. are readily available.

- Easy to index. Several spatial indexes are also provided by Oracle Spatial.

- Space saving. Simplified pipelines occupy less memory space, i.e. little redundant information is sorted.

However, there is one general disadvantage. It is possible to retrieve data from Oracle Spatial and visualize them as cylinders and rectangles in Microstation. But, the vice versa operation is not that simple. If a pipe is edited, it would be difficult (or even impossible) to estimate the change of the centreline in order to reflect the Oracle Spatial records.

Having two different views can solve this disadvantage. The first view will visualise the 3D centreline, which is basically a simple retrieval of records with geometries. This will be used for editing of pipelines information. The second view will contain the cylinder and rectangular pipelines and will be used only for 3D visualization and visual inspection. In this way, the lines will be still edited and managed as simple lines and cylinders and rectangular pipelines will be created only when needed.

3 DATA STRUCTURE OF PIPELINE

Because the information for pipelines differs per country, we have concentrated on the requirements for China as specified by the Ministry of Construction, 2005 (compare with the Netherlands in Penninga and Oosterom, 2006). The pipelines data have to include position (x, y, z), depth, size (diameter, or width and height), material, pressure, built date, number of holes, number of cables and others according to user needs. Of practical importance for our research are the parameters with spatial extend, i.e. the spatial description of centreline (x,y,z), terrain height (zt), diameter or width of ditch (d), and height of ditch (h). A point of a pipeline P can be given as a function of six parameters:

$$P = F(x,y,z,zt,d,h)\ (1)$$

However, SDO_GEOMETRY supports only 4D coordinates (Zlatanova and Stoter 2006). Since the terrain height in pipelines system is critical (i.e. used for preparing vertical profile and calculating insertion points within pipelines), we include that one as the fourth coordinate. Then, expression (1) can be modified as follows:

$$P = F(x,y,z,zt)\ \ (2)$$

From this model, the depth of pipelines can be computed as (zt-z) at a node. A pipeline is under ground if (zt-z) is positive, otherwise, the pipeline is above ground. Besides the pipeline information, very often point data are also maintained. Some points contain attribute information, which cannot be assigned to a point if it is not a separate SDO_GEOMETRY object. Practically, all the end points of a pipeline have a special meaning. They are going to be represented by a 3D symbol and therefore the coordinates (x,y,z) still have to stored in a separate point table. Additional information that helps to create the 3D symbol could be height of ground, height of pipelines bottom, azimuth, which are also to be included as attributes.

To be able to organize all these parameters in Oracle Spatial, two pre-processing steps are needed: 1) creating vertical segments and 2) defining segments as they are going to be recorded in Oracle Spatial. The original pipeline information, although with 3D coordinates, provides enough information only for 2D representations. A vertical segment is a typical example in this respect. They do not exist in the data sets as intended for 2D but they are required for 3D representations.
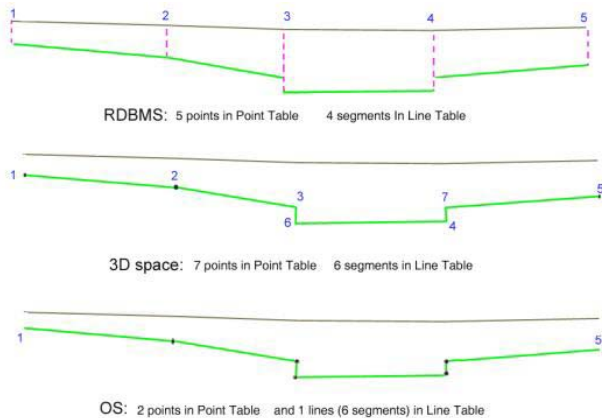

Figure 1: Conversion from 2D data to 3D

Figure 1 illustrates the transformation from the original data to the needed representation in Oracle Spatial. Starting from 5 points in the point table and 4 segments in the line table, the final representation goes to 2 points in the point tables (which contains only end-points) and 5 segments in the line table. Indeed, it should be carefully considered which and how many segments have to be organized as one line data type in the database. Very long lines might result in a weak and inefficient indexing.

4   3D VISUALIZATION IN MICROSTATION

Microstation V8 (GeoGraphics) provides means to retrieve and visualize the data types from SDO_GEOMETRY. However, this program visualizes the pipes as they are recorded in the database, i.e. as point and lines. The display of cylinder and rectangular pipelines requires the development of new programs within Microstation programming environments. There are several development environments, such as MDL (Microstation Development Language), JMDL (Java edition of MDL), VBA(Visual Basic for Applications). Below is presentation of JMDL and 3D visualization's procedures based on JMDL (see also Pu, 2005).

JMDL is a java extension of MDL. The JMDL complex types are sort of structures and static C-arrays. In the initial releases of Microstation/J, JMDL could be used to run existing Java applications, while standard Java interpreters are not able to interpret JMDL code. This is due to the built-in extensions made to the JMDL. Java can also be used to develop JMDL applications that access Microstation via the DGN Package. The DGN Package is a set of classes that allow Java/JMDL programmers to query and manipulate design (DGN) files. The methods contained in the DGN Package classes are not simply MDL functions wrapped up into JMDL methods. The connection to the Oracle is done by JDBC (java database connectivity) thin driver, a 100% Java driver for client-side use without an Oracle installation.
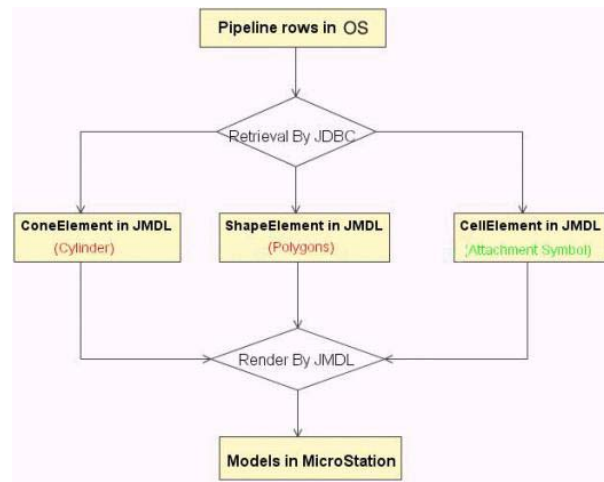

Figure 2: On-the-fly creation of 3D shapes for visualisation

Using this approach, three programs for visualization of 3D pipelines are developed (Figure 2): 1) circular pipeline, 2) rectangular pipeline, and 3) 3D symbols. The following text presents the algorithms.

4.1   3D visualization of cylindrical pipelines

3D Visualization of cylindrical pipelines is relatively simple. A cylindrical pipelines is composed of many cylinders, which can be constructed with coneElement class in JMDL according to the value of diameter and coordinates obtained from Oracle Spatial tables. These cylinders are all straight cylinders, therefore, gaps and superimposition exist at the joint between two segments (Figure 3).The size of gap may change with respect to the diameter and the turn angle of centerline. The most elegant way to solve this problem would be a torus joint, however, this shape is not available in JMDL. Another option could be a B-spline Surface (Pu, 2005). However, B-spline Surface is a rather complex shape to be used only for visualization improvements. The simplest way to close the gaps is to display a sphere at the joints of any successive segments. Sphere is the simplest surface and does not depend on the rotation matrix in 3D space. Unfortunately, sphere does not exist in JMDL class either. It was found that a cellElement class in JMDL can produce the same effect and it was used as a 3D symbol. The radius of sphere is equal to that of joining cylinders.
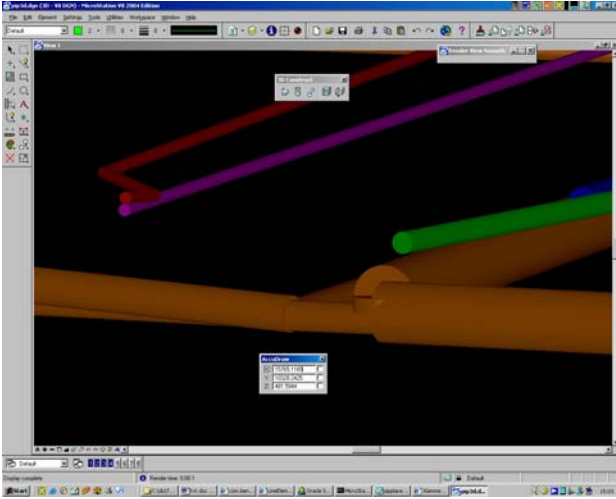
Figure 3: Gaps and superimposition at joins

Thus the 3D visualization procedure for cylindrical pipelines includes the following steps: 1) build a connection with JDBC, 2) get the diameter and coordinates value of a pipeline, 3) construct a segment (coneElement), 4) construct a sphere joint and 5) display in the view windows in Microstation. The steps 2, 3, 4 and 5 are the repeated for every line.
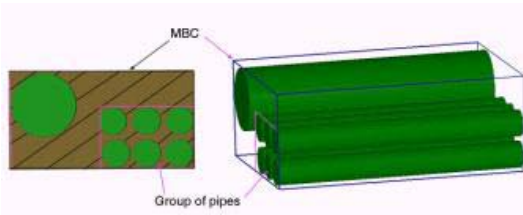

Figure 4: Rectangular pipelines as MBC of cable groups

4.2 3D visualization rectangular pipes

Rectangular pipes mainly include ditches (with cover) or the Minimum Bounding Cuboid (MBC), which minimally encloses the cables and groups of telecom and electricity, as shown in Figure 4.3D visualization of rectangular pipelines is carried out by simulating a hexahedron which is constructed by eight vertexes and four polygons. In order to reflect the visualization impression, the polygons are restricted to be planar. Considering the rectangular pipelines in the reality, the two polygons in the right- and left-hand are required to be vertical.
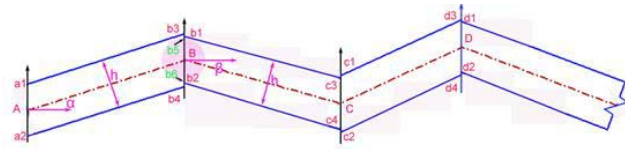

Figure 5: Section of a rectangular pipeline

The computation of vertexes of the hexahedron is the critical step of visualization of rectangular pipelines. A mathematically correct computation is quite complex but not necessary. In this paper we follow an approximated, simplified approach, which will be explained in the following text.

The coordinates of the vertexes are determined by the coordinates of the centreline and the size of the pipeline. Firstly, a vertical section is made through the centreline. In the vertical section, two lines parallel with the central line are made by two sides, respectively. The distance with the central line is half of

pipelines size (Figure 5). The cross points of parallel lines with the transects of the two ends in AB segment are presented as point a1, a2, b3 and b4. Whereas, there may be 4 points in about one central point of pipeline, i.e. b1, b2, b3 and b4 in point B. The coordinates of X, Y of the four points are the same with corresponding central point B, and Z coordinates are calculated as:

$$Z_{b1} = Z_B + h/2 / \cos(\beta)$$
$$Z_{b2} = Z_B - h/2/ \cos(\beta) \quad (3)$$
$$Z_{b3} = Z_B + h/2/ \cos(\alpha)$$
$$Z_{b4} = Z_B - h/2/ \cos(\alpha)$$

where, α and β are respectively the inclination angles of the forward and afterward segments.

If absolute value of α is equal to that of β, point b1 and point b3 are the same point, b2 and b4 are the same point too, otherwise there are always 4 points (only Z is different) at any middle vertex of centreline of pipeline, i.e. c1, c2, c3, c4 and so on. Then 3D shapes of rectangular pipelines suddenly change their height. Because the angles are rather small, $\cos(\alpha)$ and $\cos(\beta)$ are approximately equal to 1, then equation (3) can be simplified as:

$$Z_{b1} = Z_B + h/2$$
$$Z_{b2} = Z_B - h/2 \quad (4)$$
$$Z_{b3} = Z_B + h/2$$
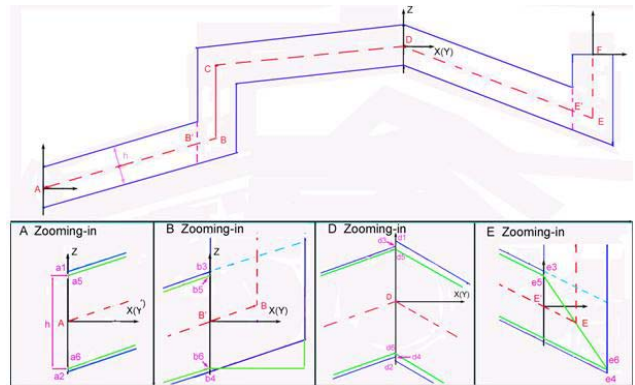$$Z_{b4} = Z_B - h/2$$

Figure 6: Four cases of rectangular intersections

After this simplification, point b1 and point b3 are moved to the location of point b5, and point b2 and b4 moved to the location of point b6 (Figure 5). In this way, we ignore the small gaps between b1-b3 and b2-b4 to ensure the 3D visualization. The disadvantage is that the height of the pipelines may be changed though the change is rather small. The computation of point 5 and point 6 may become more complicated if a pipeline has a vertical segment.

Figure 6 displays all the four possible cases of a pipeline change: 1) vertexes of the beginning (or ending) pipeline, see location A, 2) vertical middle vertexes (only Z is different), see location BC, 3) common middle vertexes, see location D, and 4) vertical ending vertexes, see location E. Below are computation expressions of point 5th and point 6th in above cases:

Vertexes of the beginning (or ending):

$$X_{a5} = X_{a6} = X_A$$
$$Y_{a5} = Y_{a6} = Y_A$$
$$Z_{a5} = Z_A + h/2$$
$$Z_{a6} = Z_A - h/2$$

Vertical middle vertexes:

$X_{b5} = X_{b6} = X_{B'} = X_B - d* Sin(A_{AB}) / 2$
$Y_{b5} = Y_{b6} = Y_{B'} = Y_B - d* Cos(A_{AB}) / 2$
$Z_{b5} = (Z_A - Z_B)* S_{B'B}/ S_{AB} + Z_B + h/2$
$Z_{b6} = (Z_A - Z_B)* S_{B'B}/ S_{AB} + Z_B - h/2$

where A_{AB} is azimuth of side AB. $S_{B'B}$ is length from B' to B. S_{AB} is length between A to B.

Common middle vertexes,

$X_{d5} = X_{d6} = X_D$
$Y_{d5} = Y_{d6} = Y_D$
$Z_{d5} = Z_D + h/2$
$Z_{d6} = Z_D - h/2$

Vertical ending vertexes

$X_{e5} = X_E - d* Sin(A_{DE}) / 2$
$Y_{e5} = Y_E - d* Cos(A_{DE}) / 2$
$Z_{e5} = (Z_D - Z_E)* S_{EE'}/ S_{DE} + Z_E + h/2$
$X_{e6} = X_E + d* Sin(A_{DE}) / 2$
$Y_{e6} = Y_E + d*Cos(A_{DE}) / 2$
$Z_{e6} = (Z_D - Z_E)* S_{EE'}/ S_{DE} + Z_E - h/2$

where A_{DE} is azimuth of side AB, $S_{E'E}$ is length from E' to E. S_{DE} is length between D to E

Secondly, from point 5th and 6th, two planes are extended according to width of the pipelines. In such a way, four points are obtained and they are combined as a vertical rectangle. These four vertexes 1,2,3 and 4, have following relationship:

$X_1 = X_4, Y_1 = Y_4, X_2 = X_3, Y_2 = Y_3$ (except Case 4)
$Z_1 = Z_2, Z_3 = Z_4$

5,6,7 and 8 are extra points for vertical middle segment. The result of the computation can be seen in Figure 7.
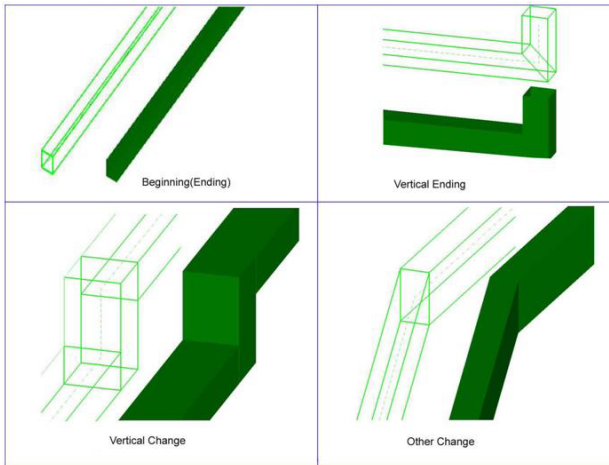

Figure 7: Rectangular pipes in 3D

4.3 3D symbols to represent connections

At certain points of the pipelines, a special kind of connections has to be visualized, such as valves, hydrants, man-made holes, etc. Generally, they are represented on 2D maps as symbols. In 3D, they have to be simulated with 3D symbols. In our case, we have created the symbol as cells in MicroStation. Below is CellElemant's construction function in JMDL:

```
CellElement(String cellstr, Dpoint cpt, dpoint
scalp, Boolean trueScale, RMatrix  rmat);
```

Where cellstr is name of cell, cpt is the location to place the cell, scalp is scale factor to be applied to the cell at directions of XYZ axis.

According to this representation, almost all information required for symbolization can be drawn from the point table. The only element still to be constructed is the 3D symbol itself. A symbol is made up of series of graphic elements and can be created as any other geometry in Microstation. Each symbol has an origin, which is defined when the cell is created. We created several basic 3D symbols, i.e. valve, hydrant, well for check, manhole, hand hole and control box, as shown in Figure 8. One can update and append a symbol at any time. In order to conveniently use symbols in the program all symbols are organised into a uniform symbol library.
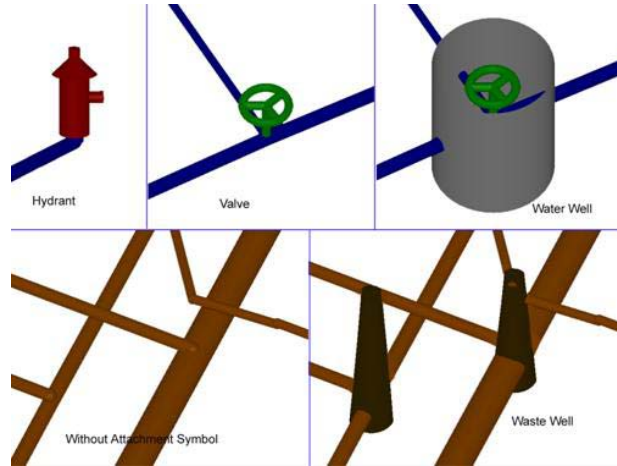

Figure 8: 3D symbols designed as cells in Microstation

5   PERFORMED TESTS

The approach was tested with data from P.R. China and the Netherlands (Chong, 2006). The utility networks were organised together (in one DBMS) with other 3D data such as terrain surface (TIN), 3D city model and cadastral parcels. These data were organised in separate tables as described by Stoter and Zlatanova, 2003. Their 3D visualisation in Microstation was completed with standard means (e.g. Spatial Viewer) to access and retrieve data from Oracle Spatial. The different data sets could be successfully integrated with the 3D representations of the pipelines and their connections. Figure 9 illustrates some of the results.
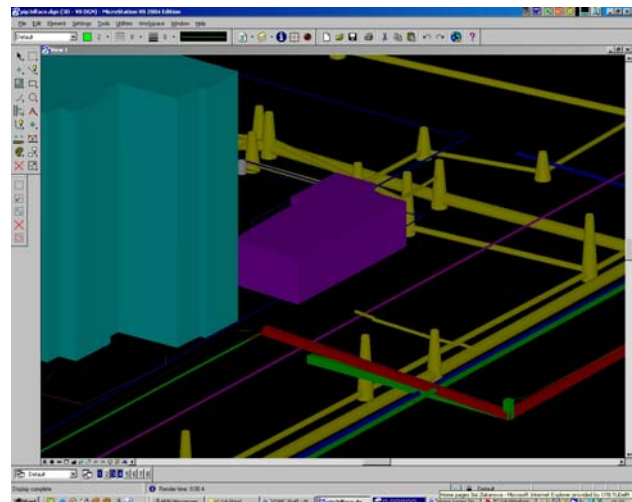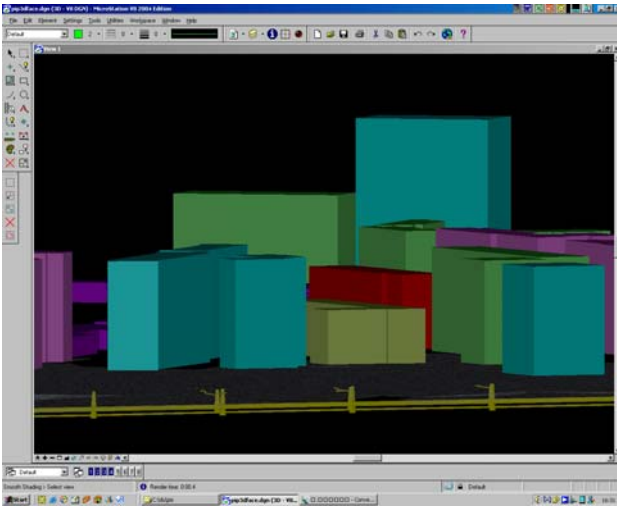
Figure 9: Snapshots of 3D visualisation of pipelines in urban environment

6   CONCLUSSION AND FURTHER RESEARCH

This paper reported our approach for creating 3D realistic visualization of urban pipelines stored as simple line data types in DBMS. All the experiments and test have clearly shown that 3D visualization of urban pipelines is better than 2D visualization. Relationships between pipes and other objects (buildings, road, terrain) are well visible in 3D scene. The chance of misunderstanding is reduced to minimum. 3D visualization of circular pipelines and rectangular pipeline, no matter underground or on the ground can be graphically represented in 3D. The inclusion of 3D symbols to show pipeline attachments (e.g. valves, hydrants, wells) helps to provide additional information regarding a particular pipeline, including its function, direction of flow, connectivity etc.

The 3D model of urban pipeline (as presented here) is a significant improvement compared to original way of data storage. Firstly, because all the vertical segments are taken into consideration. Secondly, using the SDO_GEOMETRY line data type, it was possible to reduce the number of records and thus increase the performance.

Management of the utility networks in the database allows for integrated visualization and analysis with other data sets such as cadastral and topographic data. The provided spatial functionality is still 2D but for a large number of queries this is sufficient. For example, the query 'which pipes cross a particular cadastral parcel' can be completed in 2D.

Despite the promising results, further developments are desirable at both front-end and database level. Some specific functions for pipelines, i.e. analysis of section, profiles and intersections and analysis of probably broken points, have to be developed under Microstation. Some other options to refine the 3D visualization at joins, such as torus, rectangular cylinder, sphere, etc. have to be studied. All the developments are within the old JMDL environment of Microstation V8. The new release, i.e. Microstation V8XM is object-oriented offering flexible VBA programming and extended libraries.

Further investigations are needed to specify which utility functions are worth implementing at database level. A special attention should be given to the network model since it can add value in the organization of data. Appropriate tests have to be set up to investigate the correlation between length of pipeline segments and the building of the spatial index.

We firmly believe 3D visualization of pipelines is necessary tendency of UPIS development. It can clearly express the position and spatial relationship of all pipelines, and implement arbitrary displays of pipelines from any view and/or from any place. 3D visualisation can significantly reduce blind cutting and fault damaging of pipelines.

REFERENCES

Arens, A, 2003, Modelling 3D spatial objects in a Geo-DBMS using a 3D primitives, MSc thesis, available at http://www.gmdc.nl/publications

Bentley, MicroStation/J JMDL Documentation, February 1999.

Chong SC, 2006, Registration of Wayleave (cable and pipeline) into the Dutch cadastre, Case study report, available at http://www.gdmc.nl/publications, 56 p.

GeoStar, Software System, Wuhan Geo Inc. available at: http://www. geostar.com.cn

MAPGIS, Software System, Zongdy Cyber Inc., available at http://www.mapgis.com.cn

Ministry of Construction in China, 2005, Management regulation for engineering documents of underground pipelines and cable in city, (issued in January 7th 2005) (in Chinese).

OGC, Abstract specifications overview, available at http://www.opengeospatial.org

Oracle, 2003, Spatial User's Guide and Reference10g Release 1 (10.1) Part No. B10826-01 Dec 2003

Penninga, F., 2004 Oracle 10g Topology: Testing Oracle 10g topology using cadastral data, GISt report No 26, ISSn 1569-0245, 50p.

Penninga, F. and P. van Oosterom, 2006, Kabel- en leidingnetwerken in de kadastrale registratie, GISt Report No. 42, Delft, 2006, 38 p. available at http://www.gdmc.nl/publications (in Dutch)

Pu, S., 2005, Managing Freeform Curves and Surfaces in a Spatial DBMS, MSc thesis, available at: http://www.gmdc.nl/publications

Roberts, G. et al, 2002. Look Beneath the Surface with Augmented reality, *GPS World*, February, pp. 14-20.

Stoter, J. and S. Zlatanova, 2003, Visualisation and editing of 3D objects organised in a DBMS, Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering, 22-24 January 2003, Enschede, The Netherlands, 16p

Yong, Y. 2003 Research On 3D Visualization Of Underground Pipeline, Transaction of Wuhan University, edition of science information, June, 2003

Zlatanova, S. and J. Stoter, 2006, The role of DBMS in the new generation GIS architecture, Chapter 8 in S.Rana & J. Sharma (Eds.) Frontiers of Geographic Information Technology, Springer, pp. 155-180