THE JOURNAL OF NAVIGATION (2017), **70**, 735–747. © The Royal Institute of Navigation 2017
doi:10.1017/S0373463316000898

A Unified Indexing Strategy for the Mixed Data of a Future Marine GIS

Tao Liu[1], Xie Han[1], Jie Yang[2] and Liqun Kuang[1]

[1](*School of Computer Science and Control Engineering, North University of China*)
[2](*Transportation Management College, Dalian Maritime University*)
(E-mail: dlmult@hotmail.com)

Spatial indexing technology is widely used in Geographic Information Systems (GIS) and spatial databases. As a data retrieval technology, spatial indexing is becoming increasingly important in the big-data age. The purpose of this study is to propose a unified indexing strategy for the mixed data of a future marine GIS. First, data organisation of the system is described. Second, the display condition of each type of data is introduced. These conditions are the basis for the construction of a unified indexing structure. Third, a unified indexing structure for mixed data is presented. The construction process and the search method of the indexing structure are described. Finally, we implement the indexing strategy in our system "Automotive Intelligent Chart Three-dimensional Electronic Chart Display and Information Systems" (AIC 3D ECDIS). Our strategy can provide fast and integrated data retrieval. The spatial indexing strategy we propose breaks through the limitation of data types in our system. It can also be applied in other GIS systems. With the advent of the big-data age, mixed data indexing will become more and more important.

KEYWORDS

1. Spatial indexing. 2. Data retrieval.
3. Mixed data. 4. Marine GIS.

Submitted: 17 December 2015. Accepted: 28 November 2016. First published online: 6 February 2017.

1. INTRODUCTION. With the development of spatial information applications such as Digital Earth and Digital City, there is an explosive increase in the amount of spatial data. Due to the complexity of spatial data, spatial data management and retrieval is becoming more and more difficult but they are essential for Geographic Information Systems (GIS). Efficient and precise data retrieval is a common need for GIS applications. As a key technology of spatial data retrieval, spatial indexing can greatly improve the performance of GIS data retrieval and has become an indispensable part of the retrieval function. Since the 1970s, when spatial indexing was issued, it has been widely used in spatial databases and GIS systems (Huang and Wu, 2007). Goodchild et al. (1992) emphasised the importance of spatial indexing in the development of spatial data analysis and GIS. They point out that an indexing scheme is a useful tool for handling large amounts of spatial data. In recent years,

the relative theories and technologies of spatial indexing, including indexing structure and search algorithms, have been developed extensively. This significantly improves the ability of GIS and spatial databases to process large spatial data sets. With the development of research on spatial data mining and spatial data analysis, there is an increasing demand for spatial indexing technologies. The "big-data" age further enhances this demand.

Spatial indexing structures can be broadly classified into two categories: tree structure and space mapping method. These two methods can be fused together to form new structures. The most commonly used method is tree structure. Tree structure method has two main ideas. The first idea is based on binary tree structures and the typical indexing structure based on binary tree is KD-tree (K-dimensional tree) (Zhou et al., 2008; Xu et al., 2009). This structure applies the division thought of binary tree in the K-dimensional space. KD-tree divides each dimension in proper order. It indexes the multi-dimensional spatial data by increasing the depth of the tree.

The structure is suitable for indexing point data but this structure is not suitable for indexing polygon and polyhedral data. There are many variant structures of KD-tree, such as MKD-tree, KDB-tree, and so on (Kumar, 1994; Zhang, 2011). Another structure is quad-tree (Tanin et al., 2007). This is the extension of binary tree in the two-dimensional (2D) space. The main advantage of this structure is good stability. With the insertion of data, the basic structure of this tree does not change greatly. This structure can be easily combined with other indexing methods and data access methods in the distributed environment. The main disadvantage of this structure is that it is hard to construct a balanced structure. This will affect the indexing efficiency.

The second idea is based on B-tree structure. B-tree and its variants, such as B^+-tree and B^--tree, are the most widely used indexing structures in the current computer field. B-tree is a balanced tree and has excellent query performance. Its extension structure in the spatial data field is R-tree. R-tree was proposed by Guttman in 1984 (Chávez et al., 2001). The basic idea of R-tree is to use the minimum bounding rectangle of the spatial object to describe the object and do the spatial calculations. This can greatly reduce the difficulty of spatial calculations. There are many variants of R-tree, such as R^+-tree, R^*-tree, and so on (Qing et al., 2003; Rees, 2003). Recently, for high-dimensional data, some structures based on R-tree are proposed, such as X-tree, A-tree, and so on (Kriegel et al., 1996; Sakurai et al., 2000).

Another category of spatial indexing structure is the space mapping method. There are many different variations that can be used to do the space mapping. Space-division method and space-filling curve method are commonly used methods. Grid indexing technology divides the space into equal or unequal grids. Each grid is associated with an external memory block. The grid and the external memory block can be connected by a Hash function. When the retrieval points to a grid, this technology will then load the spatial data from the corresponding external memory block. Typical grid indexing structures are Grid file and R file (Shekhar et al., 1999; Yang et al., 2005). This structure is suitable for indexing point objects. The space-filling curve method divides the space into many grids according to a division method. This method encodes each grid according to a certain rule to generate the relationship between the grid and a one-dimensional code. Spatial objects can be represented by one or more grids that intersect the object. The relationship between spatial objects can be calculated by their corresponding grid codes. This method maps the spatial object onto a one-dimensional space, so traditional indexing methods can be used to index these objects.

Spatial data are composed of different types of data, such as raster data, vector data, 3D models, and so on. Each type of data can be indexed by one type of spatial indexing method. For example, we may use R-tree to index vector data and use quad-tree to index raster data. However, few researchers have studied integrated data indexing schemes for different types of data. At present, most spatial indexing methods only deal with one type of data. Few indexing strategies are for mixed data, that is, few indexing structures can handle mixed data.

In some cases, we need integrated data retrieval. For example, in port planning, we need to query the terrain data and artificial vector data of the specified area. In some emergency cases, we need to interrogate many different types of data as quickly as possible. For example, during navigation, we have to consider Electronic Navigational Chart (ENC) vector objects, terrain data and 3D models in 3D marine GIS. All these cases need fast and integrated data retrieval. This problem will be better solved if we can provide a unified indexing strategy.

In this paper, we propose a unified indexing strategy for the mixed data of a future marine GIS. This strategy focuses on the indexing method of the server side and helps to improve function services of the server side. First, we describe the data organisation of the mixed data. Then we introduce the indexing strategy. The strategy is composed of two parts: indexing structure and storage structure. Finally, we implement our indexing strategy to provide fast and integrated data retrieval in our system "Automotive Intelligent Chart Three-dimensional Electronic Chart Display and Information Systems" (AIC 3D ECDIS). Also the application shows the effectiveness of our indexing strategy.

2. DATA ORGANISATION.

We propose a future marine GIS called AIC 3D ECDIS based on Digital Earth (Liu et al., 2014; 2015). This platform provides a 3D description of marine spatial data. It can support global spatial data and 3D visualisation and can be published on the web to provide application and data service through the network. The spatial data are mainly divided into five categories: remote sensing image data, sounding data, terrain data, 2D vector ENC data and 3D model data. All these data are fused together in a unified framework and interface. This platform solves the problem of the fusion of overland terrain and sounding values. It also solves the problem of map sheet clipping and splicing in 3D scenes according to the multi-scale attributes of the ENC data and fuses multi-scale charts into 3D scenes based on underwater topography and remote sensing images. The data fusion effects of our system are shown in Figures 1, 2 and 3.

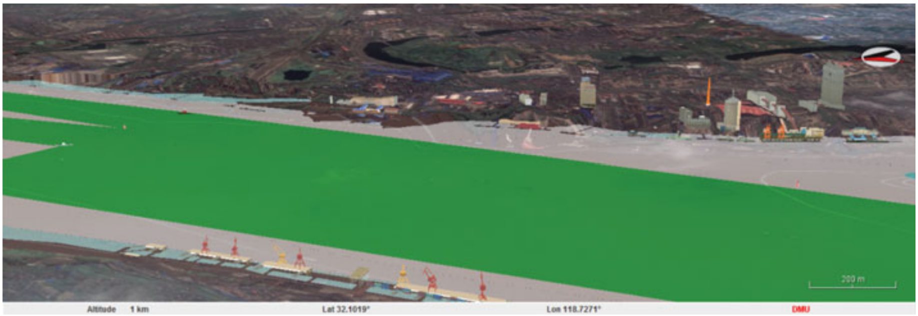


Figure 1. Overwater view effect of the fusion.

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

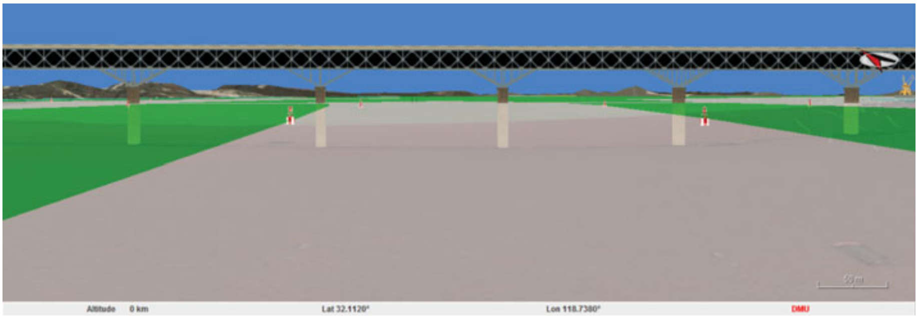


Figure 2. Water surface view effect of the fusion.

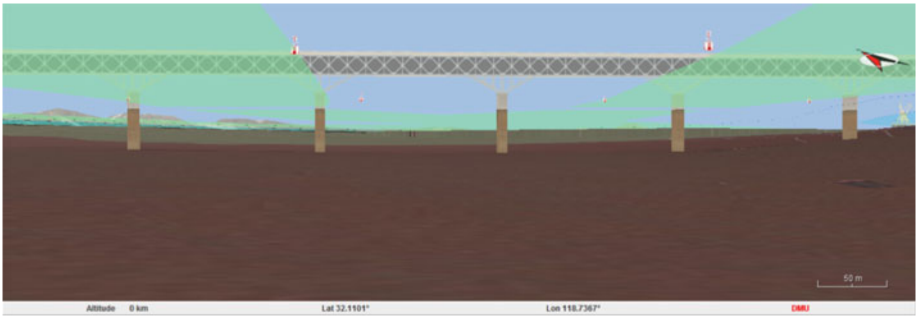


Figure 3. Underwater view effect of the fusion.

Data organisation is an important factor for the choice of indexing technology. In order to give a detailed introduction to our indexing strategy, this section first introduces the organisation of the data in our system.

2.1. *Remote Sensing Image Data.* In order to manage large amounts of remote sensing image data, we adopt the multi-resolution pyramid model to sample and organise this type of data. This multi-resolution pyramid model is a power-of-two pyramid model, that is, the lower level tile has a quarter scope and double the resolution of the parent tile. In consideration of the basic grid model (grid model of Digital Earth) of our system and the texture mapping technology of OpenGL, the pyramid model should divide the image data according to the latitude-longitude information. The top of the pyramid model is a zero level tile of $36^\circ \times 36^\circ$ extent. This size is a suitable value for global use. This model contains 12 levels, so the eleventh level tile has the exact size of $0{\cdot}017578125^\circ \times 0{\cdot}017578125^\circ$. Each level tile has 512×512 pixels. For simplicity, we use integer keys (row number and column number) to mark each tile (Liu et al., 2014). Each level is the resampling result of the original image data.

2.2. *Terrain Data.* We can process terrain data in the same way as remote sensing image data, because essentially they are both 2D data arrays. In order to efficiently use and manage large amounts of terrain data, we also adopt the multi-resolution pyramid model to sample and organise the terrain data. The terrain pyramid model has some state parameters, such as partition parameter, level parameter, and so on. In order to facilitate the indexing of the terrain data, we use the same state parameter values as the image pyramid model in our system, including the size of the zero level tile, number of levels and marking

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

rule. The reason will be explained in Section 3. Unlike the image pyramid model, each level tile of the terrain pyramid model is divided into 150 × 150 partitions to sample the terrain data. The eleventh level tile has the exact size of 0·017578125° × 0·017578125°. So the sampling interval of the eleventh level is 0·421875 arc seconds, or 1 in 150 of 0·017578125°.

We fused sounding data with overland terrain data using the distance-weighted averaging method in the pyramid model (Liu et al., 2014). This pyramid model achieves the organisation of terrain data and sounding data. When the elevation value of a certain point is required in the program, the tile containing the point will return the value by bilinear interpolation from the corresponding subdivision grid corner values.

2.3. *ENC Data.* Traditionally, ENC data are organised in the form of a map sheet. The subdivision of map sheets is irregular. Each map sheet is created based on the map sheet scale. In order to efficiently use this type of data, we use the object-oriented data model to describe each object in the internal memory. The object-oriented data model combines spatial attributes and object attributes of an object in a unified data structure and can better describe the relationship between objects. This data model meets the ENC data description specification S-100 (Alexander et al., 2007; Ward et al., 2009). In our system AIC 3D ECDIS, some ENC objects are described by 3D models, such as buoys, buildings, wharves, bridges, etc. So, we add an attribute for these objects to point to their corresponding 3D models.

2.4. *3D Model.* 3D models created by 3D modelling tools are used to describe entities. They are 3D symbols of the entities. There are two ways of organising these models and their corresponding attribute information. One is to store 3D models and their attributes in a spatial database, such as Oracle, Geodatabase of ArcGIS 10, and so on. The other is to store 3D models in file form and store their attributes in a relational database. We choose the first way. We use Geodatabase of ArcGIS 10 to store 3D models and their attributes in our system. The attributes of the 3D model include model name, model type, map sheet number, geographic position of the model centre, model elevation, model size, model rotation parameter, and so on. Model type of the 3D model is the type number of the ENC object. Map sheet number of the 3D model is the sheet number of the map sheet containing the 3D model.

3. INDEXING STRUCTURE.

Raster type data include remote sensing image data, terrain data, and so on. This type of data can be divided into pieces. Many indexing technologies can be used to index this type of data. Development and maturation of wavelet analysis and pyramid models make the indexing of this type of data simpler and easier. For vector data, there are two main organisation methods in the indexing structure: discrete object and data tile. The discrete object method can maintain the integrity of the object. The data tile method divides the vector data into pieces according to the spatial extent. Although this method can improve the efficiency of data transmission and retrieval, it undermines the integrity of topology information of vector objects. Here we choose the discrete object method. 3D models correspond to the 2D vectors by the object name, object type and map sheet number. When an object is retrieved, its corresponding 3D model can be easily retrieved from the 3D model database by the object name, object type and map sheet number.

In many cases, we need a unified indexing structure, just as described in Section 1. The unified indexing structure can save memory and reduce the number of storage and retrieval

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

operations. Data can be better organised and managed, and can be quickly indexed. Also, integrated retrieval can be better supported.

According to the data organisation of terrain data and remote sensing image data, we have to use quad-tree to index these data, while many kinds of indexing structures can be used for the ENC data. In order to obtain a unified data indexing structure, we choose quad-tree structure to index the mixed data. It should be noted that we have to solve a problem first: integrate multi-scale attributes of ENC data into the indexing structure. At present, there are mainly two solutions to construct multi-scale vector data description models: multi-scale vector data indexing technology and multi-scale vector storage structures. The commonly used multi-scale indexing technologies are Reactive-tree, GAP-tree, Multiple R-tree, and so on (Li et al., 2002; Oosterom and Schenkelaars, 1995; Kwon and Yoon, 2002). The commonly used multi-scale vector storage structures are Map Cube Model, Scale-based Model, Strip trees, BLG-tree, PR-file, and so on (Hornsby and Egenhofer (2002); Cheng et al. (2009); Bertolotto and Egenhofer (2001); Chan and Chow (2002); Jones and Ware (2005)). However, these indexing technologies and models are just for vector data. It is hard to use these methods to index raster type data. In this paper, we use quad-tree to integrate multi-scale attributes of ENC data into the indexing structure. This structure is called Raster-Vector Multi-scale Hybrid Quad-tree (RVMHQ-tree). In our system, data processing is based on the RVMHQ-tree indexing, such as data rendering, integrated data retrieval, and so on.

3.1. *Display Condition.* In order to give a detailed introduction to the indexing structure, this section first introduces the display condition of each type of data.

For the remote sensing image data, we use multi-resolution pyramid model to organise this type of data. The display condition of each image tile data is described by Equation (1).

$$\begin{cases} h < IH, \\ E(s) \cap E(t) \neq \varnothing, \\ \sqrt{x_m^2 + y_m^2 + z_m^2} \times \Delta Lat(t)/iw > d_m/10^r. \end{cases} \tag{1}$$

where h is the current height of the viewpoint, measured in metres (m); IH is the initial display height, measured in metres (m); $E(s)$ is the spatial extent of the current scene and $E(t)$ is the spatial extent of the image tile. $\Delta Lat(t)$ is the span of latitude of the image tile, measured in radian (rad); iw is the number of partitions of the image tile width; d_m is $\min(d_1, d_2, d_3, d_4, d_5)$, where d_i ($i = 1, 2, 3, 4$) is the distance between the viewpoint and the corner point of the image tile and d_5 is the distance between the viewpoint and the centre point of the tile, measured in metres (m); x_m, y_m, z_m are world coordinates of the corresponding image tile point of d_m, measured in metres (m) and r is a scale factor that specifies the approximate size of discernible lengths in the image tile relative to viewpoint distance. Its default value is 3. This means that one metre on the earth surface should be distinguishable from an altitude of 1000 metres. The bigger value will cause the applied resolution to be higher than discernible for the altitude.

This condition is used to simulate the resolution comparison of screen and image tile. If a tile meets the display condition, the tile should be displayed; otherwise the child level of the tile should be searched.

For the terrain data, we also use a multi-resolution pyramid model to organise this type of data, just as we described in Section 2.2. The display condition of each terrain tile data

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

is described by Equation (2):

$$\begin{cases} E(s) \cap E(t) \neq \varnothing, \\ \Delta Lat(t)/tw < \Delta Lat(st)/div. \end{cases} \tag{2}$$

where $E(s)$ is the spatial extent of the current scene (current grid model), $E(t)$ is the spatial extent of the terrain tile, $\Delta Lat(t)$ is the span of latitude of the terrain tile, measured in radian (rad), tw is the number of partitions of the terrain tile width, $\Delta Lat(st)$ is the latitude span of the corresponding grid of the terrain tile, measured in radian (rad) and div is the number of partitions of the corresponding grid width. This condition is used to judge if a terrain tile is suitable for providing terrain data for the current scene.

For ENC data, the data should be displayed according to the current display scale and its drawing scale. To be specific, if the display scale is bigger than 1 in 2 of the drawing scale of the object, the object should be displayed; otherwise it should not be displayed (Liu et al., 2015).

For 3D models, we adopt view-dependent visualisation technology to render this type of data. The display condition is composed of two parts: view frustum clipping and viewpoint distance judgment. If a 3D model lies inside the view frustum and its distance to the viewpoint is less than a pre-set threshold, the model should be displayed; otherwise it should not be displayed.

3.2. *Indexing Structure.* The schematic diagram for the indexing structure of RVMHQ-tree is shown in Figure 4. The left part of the figure is the meshing of the original data and the right part of the figure is the corresponding indexing structure. In this indexing structure, vector objects at different scales are inserted into different level nodes. Objects at the same sale are inserted into corresponding level nodes in clockwise order. Each node contains image tile data, terrain tile data, vector objects, and so on.

There are two main steps in the construction of the indexing structure for the mixed data:

(1) Construct the quad-tree indexing structure for the remote sensing image data and the terrain data. We use the quad-tree to index the multi-resolution pyramid model of the remote sensing image data and the terrain data. Because the multi-resolution pyramid model is a power-of-two pyramid model, it is easy to add these data subdivisions divided by the pyramid model to the corresponding nodes from top to bottom. Each node corresponds to a subdivision

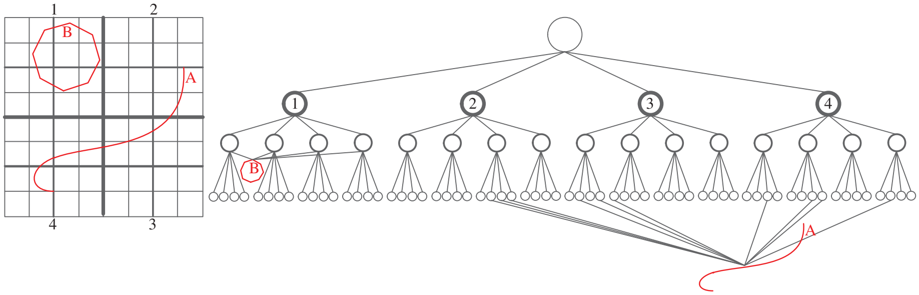


Figure 4. Indexing structure of RVMHQ-tree.

in the pyramid model and the extent of the node is the extent of the corresponding subdivision in the pyramid model.

(2) Insert ENC vector data into the quad-tree indexing structure. In order to insert the ENC vector data into the indexing structure, we should first calculate the display scale *dsl* for each level according to the display condition of the image tile data. *dsl* is calculated according to Equation (3).

$$dsl = \frac{2 \times \tan(\theta/2) \times hm}{w}. \tag{3}$$

In this equation, θ is the field-of-view angle measured in radians; *hm* is the maximum height of the viewpoint that the tile of the level should be displayed, which is measured in metres and *w* is the width of the viewport measured in pixels.

Based on the display scale of each level and the spatial extent of each node, we can insert the ENC vector data into corresponding levels according to the display condition of the ENC vector data. Firstly compare the display scale of the level and the drawing scale of the ENC object, if the display scale is no less than 1 in 2 of the drawing scale of the ENC object, stop the comparison; otherwise continue the comparison until the last level. We will get the level into which the object should be inserted. Then, calculate the minimum bounding box of the ENC object, find the nodes which intersect the object, and insert this object into the nodes' ENC object lists.

Finally, we will get the RVMHQ-tree. The indexing structure is constructed on the server side. It integrates raster type data, vector data and scale information in a unified framework. This will be very helpful for integrated data retrieval. Also this indexing structure can be combined with view-dependent Level Of Detail (LOD) visualisation technology.

This structure can provide integrated data retrievals for mixed data. In the indexing structure, ENC data are organised in the object form instead of the map sheet form. The multi-scale attribute of ENC data is integrated into the indexing structure. There are two important state parameters in a query: scale parameter and spatial extent. Scale parameter is used to determine the level of the structure and spatial extent is used to determine the branch of the structure. We will get mixed data (image data, terrain data and ENC vector data) in one query (one search of the indexing structure) according to the spatial extent, viewpoint height, meshing of the grid model and display scale. Corresponding 3D models of the ENC objects can then be retrieved from the geodatabase.

4. STORAGE STRUCTURE.

We design the storage structure of each node according to the indexing structure RVMHQ-tree described above. The storage structure of each node is described as follows:

```
typedef struct node_ RVMHQ-tree
{
   RectGeometry   range; // spatial range of the node
   int   level; // level of the node
   int   column; // column number of the node
   int   row; // row number of the node
   Child   c1; // child pointer of the node
   Child   c2;
   Child   c3;
```

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

```
    Child       c4;
    ObjectList  obj; // ENC object list in the node
    ImageTile   iT; // image tile in the node
    TerrainTile tT; // terrain tile in the node
}
```

We adopt the object-oriented data model to represent the ENC object. This storage structure can be a reference for the node structure of any mixed data indexing structure.

5. APPLICATION.

Our system AIC 3D ECDIS uses intuitive 3D models to describe entities, fuses artificial spatial objects into the 3D environment and uses 3D graphics to visualise data on the computer screen. It can provide more rich and intuitive spatial information and support more rich spatial analysis functions. Users can directly obtain ENC, riverbed, terrain and other spatial information in a unified framework and interface. Integrated spatial analysis functions can decrease mental overload and minimise fatigue. AIC 3D ECDIS will be significant for inland navigation, entering and leaving port, coastal navigation, integrated monitoring of navigation marks, and so on.

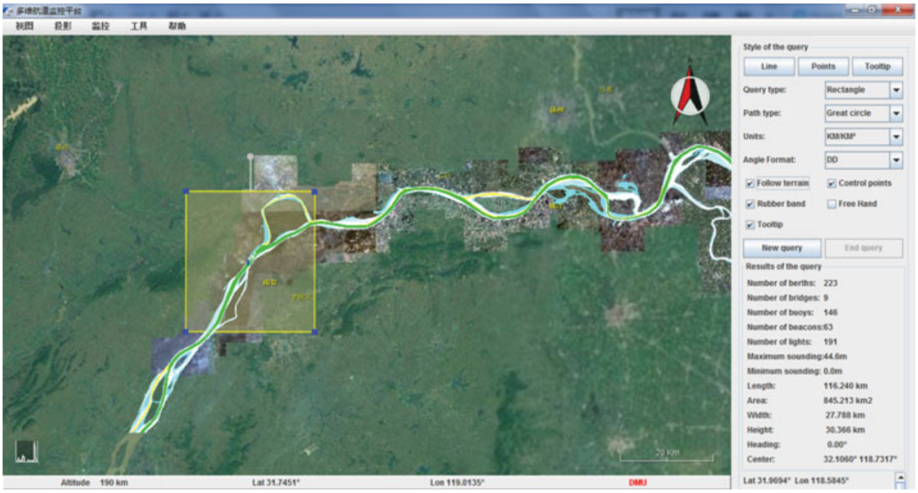
Based on the data organisation model of our system, we can use a unified data indexing strategy for mixed spatial data. From the viewpoint of fusion visualisation and integrated data retrieval, spatial data should not be divided into categories.

We integrate raster-type data, vector data and scale information in a unified indexing structure. This indexing strategy will be very helpful for data rendering and integrated data retrieval. It can support view-dependent LOD visualisation technology. Integrated data retrieval will be accomplished by one search of the indexing structure.

We implement our indexing strategy in our system AIC 3D ECDIS to provide an efficient integrated data retrieval. Also, the application shows the effectiveness of our strategy. Our indexing strategy is implemented on the server side and the indexing structure is constructed on the server side. When the client program sends an integrated query to the server side, data retrieval components in the web server will search the result through the indexing structure and the geodatabase of 3D models. Here we give some application examples in our system.

The application effect of the region query is shown in Figure 5, where the query region is marked by the yellow rectangle and the results are shown in the right panel. From this example, we can see that our indexing strategy can provide integrated retrieval results, such as number of berths, number of buoys, maximum of the sounding data, area, length, and so on.

Our system fuses sounding data with terrain data based on the pyramid model to construct a multi-resolution underwater topography. In order to intuitively describe the artificial water surface object and the underwater topography, we use the colour blend method in 3D computer graphics to draw the artificial water surface object's base a certain elevation (the elevation of chart datum, the elevation of tidal datum, the elevation of tide, and so on) on top of the underwater topography in semi-transparent mode. When the client program triggers an integrated sounding query, the result will be searched through the indexing structure. The integrated retrieval effect of sounding data is shown in Figure 6. Figure 6 shows an effect of the integrated rectangle query of sounding data at the view angle of underwater. The query region is marked in yellow, and the results are shown in the right panel.



Style of the query

Query type: Rectangle

Path type: Great circle

Units: KM/KM²

Angle Format: DD

☑ Follow terrain ☑ Control points

☑ Rubber band □ Free Hand

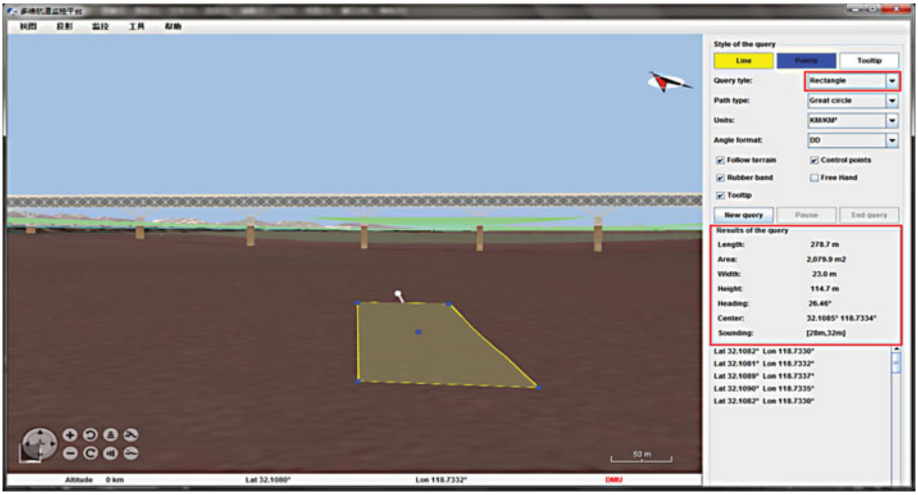
☑ Tooltip

Results of the query

Number of berths:	223
Number of bridges:	9
Number of buoys:	146
Number of beacons:	63
Number of lights:	191
Maximum sounding:	44.6m
Minimum sounding:	0.0m
Length:	116.240 km
Area:	845.213 km2
Width:	27.788 km
Height:	30.366 km
Heading:	0.00°
Center:	32.1060° 118.7317°

Lat 31.9694° Lon 118.5845°

Figure 5. Application effect of the region query.



Style of the query

Query tyle: Rectangle

Path type: Great circle

Units: KM/KM²

Angle format: DD

☑ Follow terrain ☑ Control points

☑ Rubber band □ Free Hand

☑ Tooltip

Results of the query

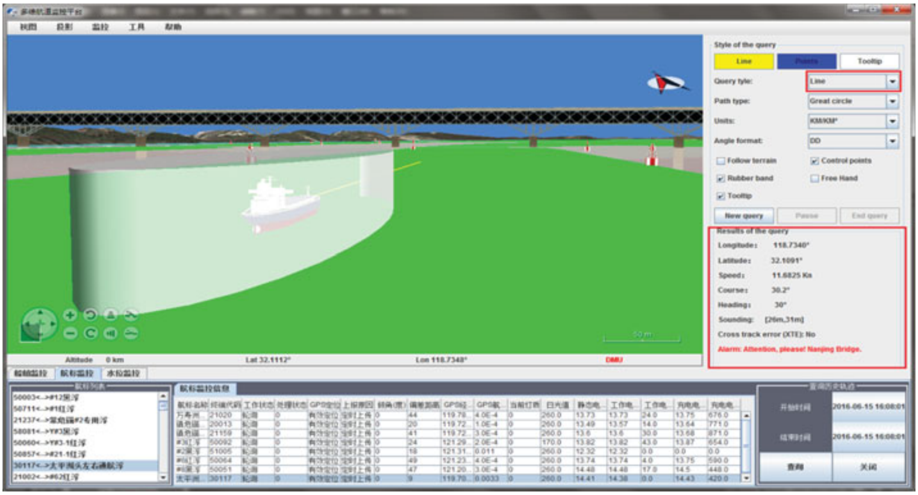
Length:	278.7 m
Area:	2,079.9 m2
Width:	23.0 m
Height:	114.7 m
Heading:	26.46°
Center:	32.1085° 118.7334°
Sounding:	[28m,32m]

Lat 32.1082°	Lon 118.7330°
Lat 32.1081°	Lon 118.7332°
Lat 32.1089°	Lon 118.7337°
Lat 32.1090°	Lon 118.7335°
Lat 32.1082°	Lon 118.7330°

Figure 6. Underwater view effect of the integrated rectangle query of sounding data.

During the navigation of own-ship, entities, such as the own-ship, navigation mark, and so on, are represented by 3D models, and their dynamic positions are monitored in real-time. When the client program triggers an integrated navigation query of the own-ship, the own-ship model will be covered by a semi-transparent ship domain elliptic cylinder or a semi-transparent distance cylinder and there is a line pointing in the heading direction. With the support of the indexing strategy of this paper, this system can provide an integrated query result by one search of the indexing structure. The result based on the forward direction, ship domain and distance threshold not only includes basic ship navigation information, but also includes sounding information, XTE (Cross-Track Error) alarm and obstacle alarm. Combined with intuitive data display, this system will make navigation more intelligent. The application effect of the integrated ship navigation query is shown in Figure 7.

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press



Style of the query

Line | Points | Tooltip

Query tyle: Line

Path type: Great circle

Units: KM/KM²

Angle format: DD

□ Follow terrain ☑ Control points

☑ Rubber band □ Free Hand

☑ Tooltip

New query | Pause | End query

Results of the query

Longitude: 118.7340°

Latitude: 32.1091°

Speed: 11.6825 Kn

Course: 30.2°

Heading: 30°

Sounding: [26m,31m]

Cross track error (XTE): No

Alarm: Attention, please! Nanjing Bridge.

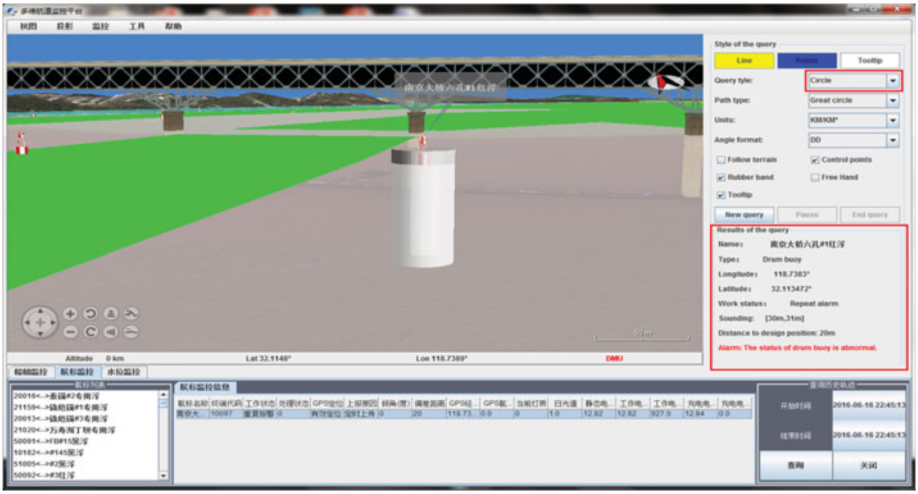
Altitude 0 km Lat 32.1112° Lon 118.7348° DMU

2016-06-15 16:08:01

2016-06-15 16:08:01

Figure 7. Water surface view effect of the integrated ship navigation query.

During the monitoring of navigation marks, dynamic positions of navigation marks are monitored in real-time. The navigation mark can be recognised very quickly from any viewpoint based on cartographic 3D visualisation. When the client program triggers an integrated query of a navigation mark, the navigation mark will be covered by a semi-transparent distance cylinder and the integrated query result will be returned by one search of the indexing structure. The result based on the distance threshold not only includes basic navigation mark information, but also includes sounding information, drift alarm, and so on. The integrated data retrieval will make the monitoring of navigation marks more intelligent and convenient. The application effect of the integrated navigation mark query is shown in Figure 8.



Style of the query

Line | Points | Tooltip

Query tyle: Circle

Path type: Great circle

Units: KM/KM²

Angle format: DD

□ Follow terrain ☑ Control points

☑ Rubber band □ Free Hand

☑ Tooltip

New query | Pause | End query

Results of the query

Name: 南京大桥六孔#1红浮

Type: Drum buoy

Longitude: 118.7383°

Latitude: 32.113472°

Work status: Repeat alarm

Sounding: [30m,31m]

Distance to design position: 20m

Alarm: The status of drum buoy is abnormal.

Altitude 0 km Lat 32.1148° Lon 118.7389° DMU

2016-06-16 22:45:13

2016-06-16 22:45:13

Figure 8. Water surface view effect of the integrated navigation mark query.

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

6. CONCLUSION. This paper provides a unified indexing strategy for the mixed data of a future marine GIS and a unified indexing structure for raster type data and vector data. We implement the strategy in our system AIC 3D ECDIS. This strategy can provide fast and integrated data retrieval. This will be greatly helpful for mixed data indexing in the age of big data.

In the integrated data retrieval case, our strategy will be very helpful. Remote sensing image data and terrain data are organised by the power-of-two pyramid model in our system so data organisation corresponds to the indexing structure. This is very helpful for data indexing and data retrieval of remote sensing image data and terrain data. The strategy integrates the multi-scale attributes of ENC data into the indexing structure. This facilitates the retrieval of ENC data and our spatial indexing strategy will be very helpful for server-side concurrency control.

At present, we have only implemented this strategy in our own system. We will implement our strategy in other GIS systems to improve the strategy in the future.

ACKNOWLEDGMENTS

We would like to thank the NASA technical team of World Wind and the contributors to the World Wind. The World Wind platform provides us with a lot of help.

FINANCIAL SUPPORT

This work was supported by the National Natural Science Foundation of China (grant number 61379080) and the Natural Science Foundation of Shanxi Province, China (grant number 201401118-3).

REFERENCES

Alexander, L., Brown, M., Greenslade, B. and Pharaoh, A. (2007). Development of IHO S-100 the new IHO geospatial standard for hydrographic data. *The International Hydrographic Review*, **8**(1), 56–66.

Bertolotto, M. and Egenhofer, M.J. (2001). Progressive transmission of vector map data over the world wide web. *GeoInformatica*, **5**(4), 345–373.

Chan, E.P.F. and Chow, K.K.W. (2002). On multi-scale display of geometric objects. *Data & Knowledge Engineering*, **40**(1), 91–119.

Chávez, E., Navarro, G., Baeza-Yates, R. and Marroquin, J.L. (2001). Searching in metric spaces. *ACM computing surveys (CSUR)*, **33**(3), 273–321.

Cheng, C., Lu, F. and Cai, J. (2009). A quantitative scale-setting approach for building multi-scale spatial databases. *Computers & Geosciences*, **35**(11), 2204–2209.

Goodchild, M., Haining, R. and Wise, S. (1992). Integrating GIS and spatial data analysis: problems and possibilities. *International journal of geographical information systems*, **6**(5), 407–423.

Hornsby, K. and Egenhofer, M.J. (2002). Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, **36**(1–2), 177–194.

Huang, B. and Wu, Q. (2007). A spatial indexing approach for high performance location based services. *Journal of Navigation*, **60**(1), 83–93.

Jones, C.B. and Ware, J.M. (2005). Map generalization in the web age. *International Journal of Geographical Information Science*, **19**(8–9), 859–870.

Kriegel, H.P., Keim, D.A. and Berchtold, S. (1996). The X-tree: An index structure for high-dimensional data. *Proceedings of the Twenty-second International Conference on Very Large Data Bases (VLDB'96)*, Mumbai (Bombay), India, 28–39.

Kumar, A. (1994). G-tree: A new data structure for organizing multidimensional data. *IEEE Transactions on Knowledge and Data Engineering*, **6**(2), 341–347.

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press

Kwon, J.H. and Yoon, Y.I. (2002). An access method for integrating multi-scale geometric data. *Advances in Databases and Information Systems: Proceedings of the 6th East European Conference, ADBIS 2002, Published in: Lecture Notes in Computer Science*, Springer: Berlin, 204–217.

Li, J., Jing, N. and Sun, M. (2002). A mechanism of implementing visualization with level of detail at multi-scale. *Journal of Software*, **13**(10), 2037–2043 [in Chinese].

Liu, T., Zhao, D. and Pan, M. (2014). Generating 3D depiction for a future ECDIS based on digital earth. *Journal of Navigation*, **67**(6), 1049–1068.

Liu, T., Zhao, D., Pan, M. and Bai, K. (2015). Fusing multiscale charts into 3D ENC systems based on underwater topography and remote sensing image. *Mathematical Problems in Engineering*, 2015.

Oosterom, P. and Schenkelaars, V. (1995). The development of an interactive multi-scale GIS. *International Journal of Geographical Information Systems*, **9**(5), 489–507.

Qing, Z., Duo, H., Xuefeng, Y. and Yeting, Z. (2003). An oracle-based data management method for large database in CyberCity GIS. *Geo-spatial Information Science*, **6**(4), 39–43.

Rees, T. (2003). C-squares, a new spatial indexing system and its applicability to the description of oceanographic datasets. *Oceanography*, **16**(1), 11–19.

Sakurai, Y., Yoshikawa, M., Uemura, S. and Kojima, H. (2000) The A-tree: An index structure for high-dimensional spaces using relative approximation. *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB'00)*, Cairo, Egypt, 5–16.

Shekhar, S., Chawla, S., Ravada, S., Fetterer, A., Liu, X. and Lu, C.T. (1999). Spatial databases-accomplishments and research needs. *IEEE Transactions on Knowledge and Data Engineering*, **11**(1), 45–55.

Tanin, E., Harwood, A. and Samet, H. (2007). Using a distributed quadtree index in peer-to-peer networks. *Vldb Journal*, **16**(2), 165–178.

Ward, R., Alexander, L. and Greenslade, B. (2009). IHO S-100: The new IHO hydrographic geospatial standard for marine data and information. *The International Hydrographic Review*, 1, 44–55.

Xu, K., Li, Y., Ju, T., Hu, S.M. and Liu, T.Q. (2009). Efficient affinity-based edit propagation using K-D tree. *ACM Transactions on Graphics (TOG)*, **28**(5), 118:1–6.

Yang, C., Wong, D.W., Yang, R., Kafatos, M. and Li, Q. (2005). Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science*, **19**(3), 319–342.

Zhang, X., Cheng, L. and Zhu, Q. (2011). Improvement of filtering algorithm for RFID middleware using KDB-tree query index. *Journal of Software*, **6**(12), 2521–2527.

Zhou, K., Hou, Q., Wang, R. and Guo, B. (2008). Real-time KD-tree construction on graphics hardware. *ACM Transactions on Graphics (TOG)*, **27**(5), 126:1–11.

https://doi.org/10.1017/S0373463316000898 Published online by Cambridge University Press