AN APPROACH TO UPDATING VECTOR FEATURES BY CLUSTERING ALGORITHM

Ding Lei[1], Huang Wei[1], Zhang Hongping[1] *, Tang Dejin[1], Zha Zhuhua[1], Zheng Xinyan[1], Wang Cong[1], Wang Zhen[1], Li Heng[1]

[1] National Geomatics Center of China, 100830 Beijing, China - (dinglei, huangwei, zhanghongping, tangdejin, zhazh, zhengxinyan, wangcong, wangzhen, liheng)@ngcc.cn

Commission IV, ICWG IV/III

KEY WORDS: GIS (Geographical information system), GIS vector data, Bounding box, Clustering algorithm.

ABSTRACT:

The availability of Geographic Information System (GIS) data has increased in recent years, as well as the need to update its data. One way of updating GIS vector data is by deleting and inserting data to the databases by manually defined unique value. Other one is by deleting and adding data manually. These methods all require manual predefined unique values or manual operations which is not suitable for updating data in a timely and fast way, especially large numbers of data are updated. Therefore, an automated vector data updating method has the potential to significantly increase productivity, particularly as existing GIS vector data application increase in size, the data become outdated more quickly. In this paper, we consider the difference between GIS vector data and other data that GIS vector data have spatial coordinates and topology relationships, propose an approach to updating of vector features based on clustering algorithm. First, the minimum bounding box and its center point of vector features are calculated, and then clusters the center points with DBSCAN to get some clusters. In the end, extract the smallest bounding box of every cluster and update vector features with the boxes. This approach uses smallest bounding box to update features is to reduce database query cost and improves the data update efficiency. Experiments show that the method is effective and feasible.

1. INTRODUCTION

Geographical Information System (GIS) is a computer assisted system for acquisition, storage, analysis and display of geographic data. Data is the core and emphasis of GIS (Han et al., 2005). The term vector data is usually used with the GIS area to refer to thematic mapping information delineated from mapping devices. For example, poi, roads, building, land, rivers, etc. These kinds of mapping features are geometrically maintained with a GIS as the shape coordinates that make up points, lines, and polygons (Doucette et al., 2009). GIS data describe a unified framework for spatial orientation and basic data for spatial analysis, especially for the vector data (Pan et al., 2014). Vector data is significant data model among current GIS data. Unlike other data, vector spatial data has a specific space reference, each feature has attribute information describing its physical and social properties and geometric information describing the location of coordinates, moreover, the geometric information imply topological relationships between features. Furthermore, the line feature and polygon feature have a minimum bounding box. Vector data applied widely to a variety of fields in recent years, such as city planning, autonomous vehicles, electronic map, metaverse, land survey and cadastral management, disaster monitoring and early warning, and so on.

Due to the rapid development of GIS application, the related GIS data is changed in moment, and the existing geographic information can not reflect the latest status, improve the update frequency and accuracy become very important (Pan et al., 2014). Current update approaches of vector data usually rely on a manual method to modify the vector features or update data based on databases technology, which query, delete and insert data by manually defined unique value. Due to human factors, this update method is prone to errors, which are not easily detected, very easily affected by operator capabilities. For example, the unique value defined manually may be duplicate. In addition, data update method based on unique values is a general databases update method, and there may be a new one based on the characteristics of spatial data.

However, the vector data update quickly and timely is emphasis and also challenges. Manual methods are always long update time, low update efficiency, data redundancy. Furthermore, the efficiency of manually updated data is easily affected by the operator's ability, these may be cannot ensure data updates and application on time.

Aiming to overcome these drawbacks, this study proposes an approach to updating of vector features based on clustering algorithm. The approach is based on the bounding box which is important characteristics of vector data. First, Calculated the bounding box of line features and point features, and then calculate the coordinates of the center point of the bounding boxes. For point feature, its center point is itself. Second, cluster these center points using the DBSCAN (density based spatial clustering of applications with noise) (Ester et al., 1996). After clustering, center points are divided into several clusters. A cluster is a collection of center points and there is at least a center point in the cluster. Third, according to the vector features corresponding to the center points, extract the smallest bounding box of each cluster. In the end, we use experiments show that the method uses the smallest bounding box to extract the data update area more accurately, without redundancy, and can update vector data without artificially defined unique values, which improves the data update efficiency.

* Corresponding author

2. CHARACTERISTICS OF VECTOR DATA

Vector data are widely used in a variety of applications such as mobile navigation, autonomous driving and city planning. 2D GIS models data can be divided into two main models, raster data model and vector data model. This paper focused on the vector format of GIS data. The vector data model is a significant part of the 2D spatial data model that is a common graphical data structure used primarily to represent the interrelationships among the geometric features of map graphic data and between them and attribute data. Expressed spatial entities by recording coordinates. Therefore, vector data can more accurately determine the spatial location of entities.2D space vector data included three feature classes: point, line, and polygon, which represent point features, line features, and surface features respectively. There was an ordered set of discrete point plane coordinates representing the map graphic which include point, line and polygon. Point feature can be described by a coordinate value (x, y). Line (or link) feature was available to describe a sequence of coordinates ((x_1, y_1), (x_2, y_2), (x_3, y_3) …). The sequence had at least two coordinate points. It can be used to represent linear features of straight line or curve such as, traffic line, river line, boundary, etc. Polygon (or area) feature can be described by a sequence of coordinates ((x_1, y_1), (x_2, y_2), (x_3, y_3) … (x_1, y_1)) which start point and end point is closed. The sequence was essentially a closed line and was usually used to represent polygon feature, such as building, land, lake, etc.

There is a particular property of vector data called a bounding box. Bounding box is a rectangle formed by the minimum and maximum coordinates in both horizontal and vertical axes of vector feature (Abubahia et al., 2004). There are many advantages in the bounding box. First, it is easy to count the spatial extent. Second, it has only four points, no matter how many nodes the feature has. Third, it is easy to data analysis and mining and reduced computational complexity. Fortunately, some vector data formats interested bounding box as a property, such as the ESRI shapefiles.

3. CLUSTERING ALGORITHM

Clustering is a process of grouping a set of physical or abstract objects into clusters of similar objects. The cluster is a collection of data objects that are similar to the objects within the same cluster and are dissimilar to the objects in other clusters. Although classification is an effective means for distinguishing groups or classes of objects, it often requires costly collection and label of a large set of training tuples or patterns, which the classifier uses to model each group. In contrast, clustering does not require such label at all (Han et al., 2009). Clustering is an unsupervised classification technique which does not have predefined labelled data. Cluster analysis has been widely used in numerous applications, including pattern recognition, data analysis, and image processing. Many scholars have proposed several useful and popular spatial data clustering algorithms in the past decade. Many study areas are using many varied kind of clustering algorithms to separate datasets into groups and have good quality results nowadays (Nagpal et al., 2013). DBSCAN is one of them, which can discover clusters of any arbitrary shape and can handle the noise points effectively (Borah et al., 2004).
DBSCAN (density-based spatial clustering of applications with noise) is a density-based clustering algorithm. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databasess with noise. It defines a cluster as a maximal set of density connected points. It is a process of partitioning or outlier detection to find an objects. Compared with the traditional K-Means algorithm, the most significant advantage of DBSCAN is that it does not to input the number of categories.

4. METHODOLOGY

4.1 Workflow

The workflow of the proposed approach, which is illustrated in Figure 1, consists mainly of three stages: data processing, features clustering, and bounding box building. First, we prepared the vector data to be updated, including one or more of the three feature classes of point, line and polygon. We then extracted all the needed spatial information of each vector feature according to the definition of index presented in Section 4.2, such as the minimum bounding box of each vector feature, center point of the bounding box. The next step was to cluster vector features. We used a clustering algorithm to divide vector features into several bunches. There was at least one vector feature in every bunch. Specifically, suitable objective function and distance threshold must need to be used in the algorithm. In the end, we calculated the minimum bounding box of each bunch. Spatial range information was implied in the bounding box of bunch that had minimum coordinate and maximum coordinate. And then updated the data by determining a rectangular range from the coordinates.

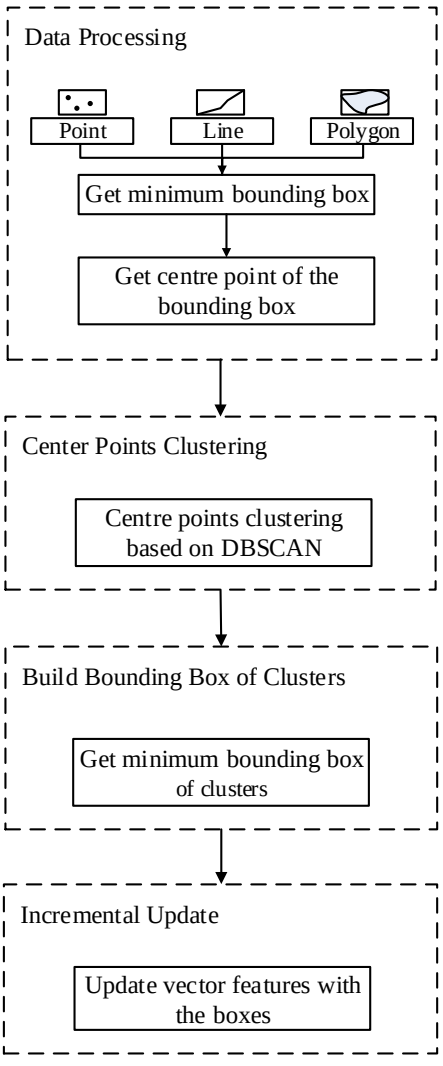
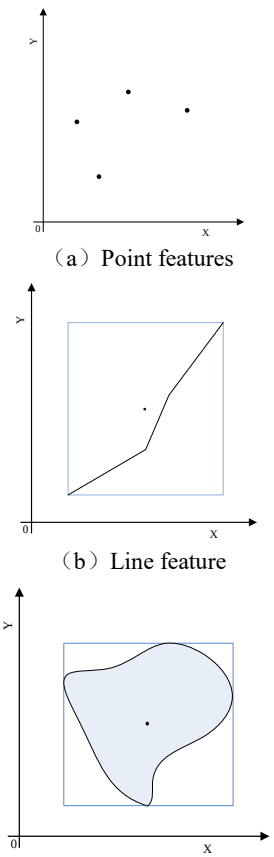


Figure 1. Workflow.

4.2 Data Processing

In the proposed approach, the GIS vector data that have coordinates and attributes has been used for processing. A spatial feature may have many nodes, such as sinuous river line or irregular building polygon, that increase computational complexity and reduces computational performance. To address the issue, this study proposed a data processing method that extract the bounding box (the minimum and maximum coordinates) and its center. The method avoided the original spatial features that had large nodes. The GIS vector datasets for processing had one or more feature classes which include point (as shown the point feature in Figure. (2a)), line (as shown the line feature in Figure. (2b)) and polygon (as shown the polygon feature in Figure. (2c)) class. The bounding box is a rectangle which contains the minimum and maximum vertex coordinates in both horizontal and vertical axes respectively. The box is also the smallest rectangle that the vector feature is ringed. It would be emphasized that all rectangles must be in both horizontal and vertical axes of the unified coordinate system. As shown the rectangles in Figure. (2). Then, the center of each bounding box also may be calculated. The x and y coordinates of center are half the minimum and maximum coordinates in both x and y axes respectively, as shown in Equation (1) and Equation (2). As shown the center points in Figure. (2). For a point feature, its center is itself. And its bounding box is none. Both line feature and polygon feature all have the bounding box properties, and easy to get its center coordinates. Finally, the data processing was ready.



（a）Point features

（b）Line feature

（c）Polygon feature

Figure 2. Bounding boxes and its centers.

The coordinates of bounding box center is given as:

$$x_i = \frac{x_{min}+x_{max}}{2}, \qquad (1)$$

$$y_i = \frac{y_{min}+y_{max}}{2}, \qquad (2)$$

where x_i = the coordinate of bounding box center in x axes
y_i = the coordinate of bounding box center in y axes
x_{min} = the minimum vertex coordinate in x-axis
x_{max} = is the maximum vertex coordinate in x-axis
y_{min} = the minimum vertex coordinate in y-axis
y_{max} = the maximum vertex coordinate in y-axis

4.3 Center Points Clustering

The approach presented in this paper was based on DBSCAN. The bounding box centers were used as the input datasets for the algorithm. GIS vector data are distributed randomly, and not known to be divided into several categories. Therefore, the DBSCAN algorithm is suitable for solving this issue because it does not need to set the number of clustering. There are many important parameters in the algorithm. The ∈ is a radius of a given object as shown in Figure. (3). The objects that distance value between an object and other objects is less than the ∈ called the neighbourhood as shown the dashed circle in Figure. (3). The white point is called seed point of the neighbourhood and the black point is called remaining point in Figure. (3). The ***MinPts*** is at least a minimum number of objects that the neighbourhood contains. The number of objects in each clustering must be greater than or equal to ***MinPts***. In this method, there is at least one vector element in each clustering (***MinPts***=1). Another important question is how to measure the nearest object. The objective function is used based on the euclidean distance, as shown in Equation (3). The objective function is a method that defined how to measure distance between two objects.
The objective function of the euclidean distance algorithm is given as:

$$L_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \qquad (3)$$

where L_i = the euclidean distance length
x = the seed point coordinate in x-axis
y = the seed point coordinate in y-axis
x_i = the remaining point coordinates in x-axis
y_i = the remaining point coordinates in y-axis

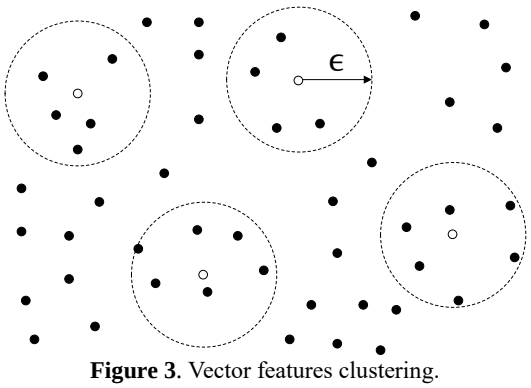


Figure 3. Vector features clustering.

How did DBSCAN find vector features clusters? The algorithm is given as Table 1. In the algorithm, the center point of bounding boxes were used as input datasets $\boldsymbol{D} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_i}\}$. The $\boldsymbol{p}$ is the center point of bounding box. A center point maps a vector feature. The ∈ is a distance length threshold that was used to find remaining points that distance length is less than threshold to the seed point for a cluster as show Figure. (3). The distance is euclidean distance. Next, the remaining points continue to repeat the process until all points are in one cluster. Each point is randomly selected by this recursive process to be clustered. Every cluster has one point at least, and a point belongs to one cluster

at most. The output of algorithm is a set of clusters that contains one and more points.

Algorithm 1: Cluster *of Vector Features*

Input: Vector features $\boldsymbol{D} = \{\boldsymbol{d_1}, \mathbf{d}, \dots, \boldsymbol{d_i}\}$, Distance length threshold ∈
Output: Clusters $\boldsymbol{C}$
1: Repeat the following steps until all points were clustered
2: Selected randomly a seed point of the unclustered points and calculated the distance length from the point to others
3: Record the points $\boldsymbol{c_k}$ that distance length is less than the threshold ∈;
4: **Return** Cluster $\boldsymbol{C} = \{\boldsymbol{c_1}, \boldsymbol{c_2}, \dots, \boldsymbol{c_k}\}$

Table 1. Clustering algorithm

4.4 Build Bounding Box of Clusters

The main purpose of this paper was to find a suitable spatial area to update vector data with the area. These areas were the extents of these clusters. So the problem was to calculate the extent of the clusters. As show in Figure. (4), every vector feature had a minimum bounding box (blue rectangle in Figure. 4) for line and polygon feature in the cluster, point feature had no minimum bounding box. The cluster contained a number of minimum bounding boxes. The minimum bounding box of feature had minimum coordinates and maximum coordinates. These coordinate values were the key basis for constructing the minimum bounding boxes of the cluster. As show in Equation (4-7), the minimum coordinate values (x_{min}, y_{min}) of cluster is the smallest coordinates (x_{imin}, y_{imin}) among minimum bounding box of features and point feature coordinates. And the maximum coordinate values (x_{max}, y_{max}) of cluster is the biggest coordinates (x_{imax}, y_{imax}) among minimum bounding box of features and point features coordinates. A bounding box can be obtained by both maximum and minimum coordinates. As show in Figure. 4, the black rectangle was the bounding box of cluster also is minimum box. After the vector elements were clustered, multiple clusters were formed. A bounding box represented an extent, and a group of bounding box represented multiple extents. These extents are used for vector data update by the spatial area.

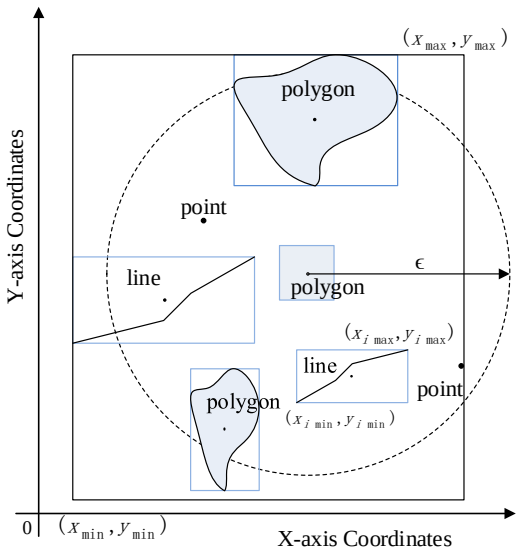


Figure 4. Vector features in a cluster.

The minimum coordinates of bounding box of the cluster are given as:

$$x_{min} = min(x_{1min,}x_{2min}, \dots x_{imin}), \quad (4)$$
$$y_{min} = mi\,n(y_{1min,}y_{2min}, \dots y_{imin}), \quad (5)$$

The maximum coordinates of bounding box of the cluster are given as:

$$x_{max} = ma\,x(x_{1max,}x_{2max}, \dots x_{imax}), \quad (6)$$
$$y_{max} = max(y_{1max,}y_{2max}, \dots y_{imax}), \quad (7)$$

where $\boldsymbol{x_{imin}}$ = minimum x coordinate of the $\boldsymbol{i}$ bounding box
$\boldsymbol{y_{imin}}$ = minimum y coordinate of the $\boldsymbol{i}$ bounding box
$\boldsymbol{x_{imax}}$ = maximum x coordinate of the $\boldsymbol{i}$ bounding box
$\boldsymbol{y_{imax}}$ = maximum y coordinate of the $\boldsymbol{i}$ bounding box
$\boldsymbol{x_{min}}$ = the minimum x coordinate of the cluster
$\boldsymbol{y_{min}}$ = the minimum y coordinate of the cluster
$\boldsymbol{x_{max}}$ = the maximum x coordinate of the cluster
$\boldsymbol{y_{max}}$ = the maximum y coordinate of the cluster

4.5 Incremental Update

In the paper, we use the PostgreSQL databases and PostGIS extension. PostgreSQL is an open source relational databases management system, it is arguably also the most advanced, with a wide range of features that challenge even many closed-source databases (Drake et al., 2002). PostGIS is the spatial databases extension to PostgreSQL databases. PostGIS provides spatial objects for the PostgreSQL databases, allowing storage and query of information about location and mapping. PostGIS adds geometry data types and spatial functions to the PostgreSQL database. The supported geometry data types are “Points”, “LineStrings”, “Polygons”. Spatial functions enable the analysis and processing of GIS objects. Examples are measurement functions like “Area”, “Distance”, and “Perimeter” and spatial operators like “Union”, “Difference”, and “Buffer”, and topological relationships like “Equals”, “Intersects”, “Crosses”, “Within”, “Contains” and “Overlaps” (Strobl, 2008).

The bounding box of clusters provided coordinates describing its spatial location. This coordinate ranges can be used for database-based data incremental updates. In the incremental update, the data to be updated is in a new table. For example, the “new_buildings” is a new table, the “gis_osm_buildings” is primary table. The process is as follow.

- Delete the exiting features in the primary table according to the update extents as follow sql:

```
DELETE FROM gis_osm_buildings WHERE shape &&
st_transform (
  st_geomfromtext (
    st_astext (
      st_makeenvelope ( x_min, y_min,
                        x_max, y_max, 4326)
      )
  ,4326)
,3857)
```

- Insert the new data in the primary table as follow sql:

```
INSERT INTO "public"."gis_osm_buildings"("osm_id",
"code", "fclass", "name", "type", "shape")
SELECT * FROM DBLINK ('new_buildings','select
"osm_id", "code", "fclass", "name", "type", "shape" from
new_buildings')
AS T("osm_id" varchar(10), "code" int2(16), "fclass"
varchar(28), "name" varchar(100), "type" varchar(20),
"shape" "public"."geometry");
```

5. IMPLEMENTATION

5.1 Data Processed

An experiment was implemented to assess the effectiveness and feasibility achieved by our proposed approach. The experimental datasets for incremental update in this paper were obtained from Open Street Map (OSM). The area of the experiment datasets was a resident area of 1.5 square kilometres. As show in Figure. 5, the datasets included poi points, road lines, and building areas, which represent point features, line features, and polygon features respectively. These features were in the mercator coordinate system WGS 84.

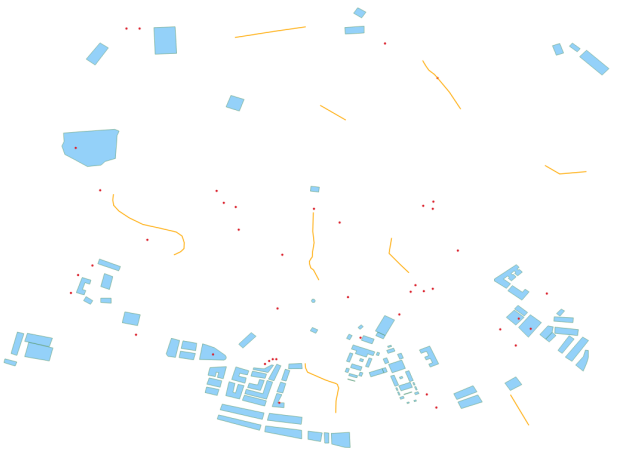


Figure 5. GIS vector map datasets.

The experiment was carried out on GIS vector maps of 43 points, 9 lines and 101 polygons, as show in Table. 2. Although the amount of data was not large, it was enough to prove the effectiveness of this approach.

Feature class	Count
Point	43
Line	9
Polygon	101

Table 2. Count of vector features

5.2 Cluster Parameter Selection

There were many important parameters in the DBSCAN algorithm. The value of $\in$ and *MinPts* were given as shown in Table. 3. In this experiment, due to every center point belongs to a cluster, the value of *MinPts* was 1. The value of $\in$ was empirically defined based on expert knowledge that considered the update extent and the density of the vector features. The value of $\in$ was 10 m.

Parameter	Value
$\in$	10 m
MinPts	1

Table 3. Clustering parameter

5.3 Results

The experimental results are show in Figure. (6), We can see that these 153 vector features were distributed among 35 bounding box of clusters. The results suggested that (1) The bounding box of clusters had a vector feature at least, every cluster had vector feature, vector features are not lost. (2) Features of different feature classes may be located in the same bounding box, the bounding box is the minimum box of them. The method based on the center point clustering can be compatible with various classes of features The coordinates of bounding boxes were as show in Table. 4. This paper listed minimum and maximum coordinates of six bounding boxes. The data were updated according to the range determined through these coordinates.

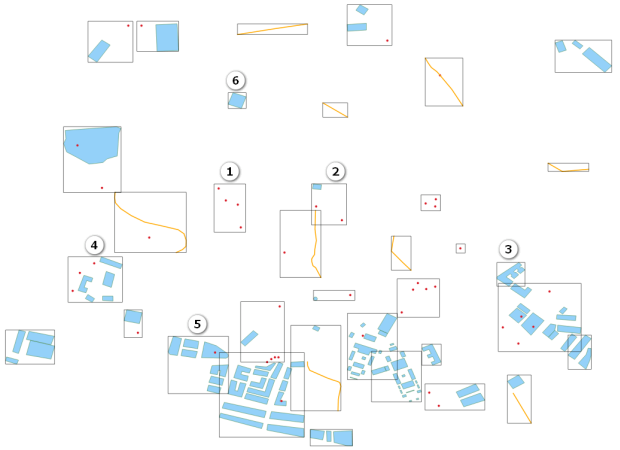


Figure 6. Bounding box of clusters.

Bounding box of Clusters	Minimum coordinates	Maximum coordinates
01	104.996474 29.184842	104.997166 29.185901
02	104.998612 29.185003	104.999385 29.185903
03	105.002684 29.183656	105.003302 29.184178
04	104.993267 29.183304	104.994459 29.184302
05	104.995464 29.181299	104.996789 29.182559
06	104.996781 29.187559	104.997178 29.187903
⋮	⋮	⋮

Table 4. The Coordinates of bounding box

5.4 Discussion

In this experiment, we extracted the smallest bounding box of vector features to be updated using a clustering algorithm. Clustering with the bounding box center of vector features avoided the complex topological relationship, such as a polygon with holes inside, reduced compute complexity. This experiment utilized the smallest bounding box to incrementally update the vector features. Because of the incremental update, the existing data inside this box must be replaced. Further, the minimum bounding boxes provided accurate areas for updating, and can save costs for spatial query operation. It is important that vector features are updated with promptly and accurately in GIS.

6. CONCLUSIONS

Although GIS technology has powerful ability of analysing and managing spatial data, updating data is problem. Aiming to overcome vector data update drawbacks, this study proposes an approach to update vector features based on clustering algorithm. The contribution of this study is hence two-fold: (1) this approach use extent to update vector data without artificially defined unique values in database, which improves the data update efficiency. (2) this method based on smallest bounding box reduce database query cost, and extents is accurate compared to larger boxes.

ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China (NO.2017YFB0503705) and the National Geo-information Service Platform "Tianditu".

REFERENCES

Abubahia, A., & Cocea, M. (2015, June). A clustering approach for protecting GIS vector data. In *International Conference on Advanced Information Systems Engineering* (pp. 133-147). Springer, Cham.

Borah, B., & Bhattacharyya, D. K. (2004, January). An improved sampling-based DBSCAN for large spatial databasess. *In International conference on intelligent sensing and information processing, 2004. proceedings of* (pp. 92-96). IEEE.

Doucette, P., Kovalerchuk, B., Kovalerchuk, M., & Brigantic, R. (2009). An evaluation methodology for vector data updating. *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, 7334, 73341F.

Drake, J. D., & Worsley, J. C. (2002). *Practical PostgreSQL*. " O'Reilly Media, Inc.".

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). *A density-based algorithm for discovering clusters in large spatial databasess with noise*. In kdd (Vol. 96, No. 34, pp. 226-231).

Goodchild, M. F., Longley, P. A., Maguire, D. J., & Rhind, D. W. (2005). Geographic information systems and science. *Wiley & Sons, West Sussex, UK*, 17, 517.

Han, J., Lee, J. G., & Kamber, M. (2009). An overview of clustering methods in geographic data analysis. *In Geographic Data Mining and Knowledge Discovery, Second Edition (pp. 149-188).* CRC Press. https://doi.org/10.1201/9781420073980

Han, M., Tian, X., & Xu, S. (2005, July). Research on data collection and databases update of GIS based on GPS technology. *In Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005*. IGARSS'05. (Vol. 2, pp. 4-pp). IEEE.

Hu, X., Ding, L., Shang, J., Fan, H., Novack, T., Noskov, A., & Zipf, A. (2020). Data-driven approach to learning salience models of indoor landmarks by using genetic programming. *International Journal of Digital Earth*, 13(11), 1230-1257.

Pan, J. P., Xu, Q. L., & Yang, C. H. (2014). Research and Application of GIS Data Update Technology. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(4), 195.

Peng Zhang, Xiaogang Wang & Peter X.-K Song (2006) Clustering Categorical Data Based on Distance Vectors. *Journal of the American Statistical Association*, 101:473, 355-367, DOI: 10.1198/016214505000000312

Nagpal, A., Jatain, A., & Gaur, D. (2013, April). Review based on data clustering algorithms. *In 2013 IEEE conference on information & communication technologies* (pp. 298-303). IEEE.

Strobl, Christian (2008) PostGIS. *In: Springer. pp. 891-898. ISBN 978-0-387-30858-6.*