A Spatio-Temporal Geocoding Model for Vector Data Integration

Xiaojing Yao(✉), Ling Peng, and Tianhe Chi

Institute of Remote Sensing and Digital Earth,
Chinese Academy of Sciences, Beijing 100101, China
yaoxj@radi.ac.cn

Abstract. Vector data integration is an important function in Urban Public Participation GIS Platform (UPPGP). Most current researches drill down the issue without considering two points: (1) the inner connections among different urban elements. (2) The temporal meaning of each object. The neglect of these points causes redundant storage and inefficient retrieval problems in smart city applications. In view of that, a spatio-temporal geocoding model for vector data integration is proposed in this paper. The model regards the urban entity element as the bridge between economic element and event element, so the task turns to find a way to uniquely identify the urban entities to avoid ambiguity and redundancy when entity objects connect with other type of objects during integration. Based on the object-oriented spatio-temporal data model, the entity object is constructed by type, space and time codes using concept lattice and regional GeoHash technologies. The method computes code similarity for each entity object to decide whether to put the object into storage. Experiments on the real UPPGP of Sino-Singapore Tianjin Eco-city show that it can avoid data redundancy and ambiguity effectively.

Keywords: Geocoding · Concept lattices · Ontology · GeoHash

1 Introduction

Big data issue is leading a new intelligent revolution in recent years. Geographic information systems (GIS) have sprout up continually with applications that include infrastructure maintenance, resource management, planning et al. But a lack of standards leads to a general inability for one GIS to interoperate with another [1]. It accelerates the production of Urban Public Participation GIS Platform (UPPGP). Since 2008, enterprise forerunners leading by IBM have formed a series of UPPGPs based on data integration and sharing [2]. The platform has many functions containing data extraction, cleaning and integration. Among these operations, integration for the vector data is one of the most important steps. It is a process of arranging the multi-source vector data in the geographical frame actually.

Multi-source vector data are different in terms of semantic meaning, spatio-temporal status, gathering approach, attribute structure et al. There are 3 integration levels of vector data currently: (1) Level 1 is a loose way of unifying the vector data in the same coordinate framework. This level refers to some basic pre-processes before deeper

F. Bian and Y. Xie (Eds.): GRMSE 2015, CCIS 569, pp. 566–577, 2016.
DOI: 10.1007/978-3-662-49155-3_59

integration, such as coordinate transformation and overlaying approaches [3, 4]. (2) Level 2 is to build relationships between individual objects in different data sets explicitly [5]. Compared with level 1, level 2 is a more compressed approach for integration. It is related with some data interoperation and information fusion technology, such as data warehouse and spatial "Extract-Transform-Load (ETL)" [1, 6, 7]. (3) The top level is the true integration level including two aspects: semantic integration and geometric integration. The former integration is related to the common attributes of the data objects. Based on the attributes belonged to different data, researchers proposed a valid method called "Geographic ontology" to identify the similarities and heterogeneities between geographic categories. Geographic ontology is a theory of abstracting entities from geographic knowledge, information and data. It makes up a system mixed with certain relationships by defining entities conceptually and formally [8]. Numerous attempts have been carried out to deal with geographic ontology integration. Among them, Kokla et al. (2001) and Kang et al. (2012) use ontology and concept lattice for geographical classification [9, 10], which provides gist for our researches. The latter integration refers to the geographic matching technology of vector data sharing the same name. Some studies have been proposed based on the type of the vector data [11] or the topology of objects [12].

The top level integration can compress several vector datasets into one really. However, most current studies concentrate on the semantic and geographic integration separately, and barely consider two points: (1) the connections among different urban elements. Urban data contain entity and event elements. Generally, there are always M: N relationships between entities and events. Taking building data for example, the management departments and the planning departments have distinct descriptions about the same building. Even more, different buildings may have the same event data also. The classical integration methods cannot describe the relationships of single building, building area, and different events happened on these individuals, because these data items have different spatial and time granularities. (2) The temporal meaning of each object. Most integration processes emphasize more on the spatial and sematic meaning than the temporality of single object. Actually, vector data from different departments always show the changes of objects along time axis. Without the time constraint, the data would lose their meanings. To sum up, the two points cause redundant storage and inefficiency retrieval in practical applications really.

Owing to that, by using geographic ontology classification and geocoding technologies, this paper proposes a spatio- temporal geocoding model for vector data, which is composed of type, space and time codes. It's a transition level between the existing level-2 and the level-3 integration, as well as a compensating model based on the content and logical connections of urban elements, and lets the disordered urban data expand infinitely and orderly in the geographical framework. A series of applications and experiments on the real UPPGP of Sino-Singapore Tianjin Eco-city show that the model can avoid data redundancy and ambiguity effectively.

The advantages of our model are summarized into threefold: first, it is excellent robustness. The granularity of property and spatial attributes of object will not break the coding framework. We can expand the code length to express more detailed information about the object. Second, it can avoid the redundant storage remarkably, because a geocoding method is proposed to compute the coding similarity for each

vector object to decide its entering. Third, the code isn't an identity authentication only, since the code itself contains information about type, space and time of an entity. By this code, irrelevant instances can be filtered out firstly, which can improve the whole efficiency of data analysis.

The paper is organized as follows. We present the constitution and logic connection of urban elements firstly in Sect. 2. Section 3 presents the integrated coding framework of urban entity elements. Then we give the implement of the urban entity coding algorithm in Sect. 4. Section 5 presents applications and experiments to demonstrate the improvements brought by the coding method for the real UPPGP. Finally, the study is summarized in Sect. 6.

2 The Construction and Logic Connection of Urban Elements

Urban is an integrated and self-operating system by combining different social and nature spaces together. To describe the variety of urban data, urban planners divide the urban data into entity elements, economic elements and event elements. Entity elements are the general terms of artificial elements and the nature elements involved by human, such as urban zone, building, traffic, water conservancy, urban pipeline. Economic elements are financial organisms that have free behavior and independent responsibility, such as government, company and human. Event elements are the sum of the activities that entity and economic elements involved in, such as urban security, urban management, and urban operation [13]. The instance belonging to these elements are called "object" in the following.

There are two types of spatial-related events in urban: one represents the status of entity objects along the time axis, such as the natural damages of streetlights. The other represents the status of economic objects depending on entity objects along the time axis, such as workers report the broken streetlights. The main difference of the two is: the former is a description of entity objects without the participation of economic objects; the latter is a description of economic objects with the participation of entity objects. That is to say, the entity elements give space supplies for the event elements. From the previous analysis, entity element is the link between economic element and event element. They have complex dependencies in time and space scales. Figure 1 is an illustration about it.

(1) Spatial dependencies. Entity elements are the spatial carriers of event and economic elements. In other words, event elements and economic elements depend on entity elements in spatial terms. As shown in Fig. 1, event elements, such as macro economy, region planning and street management, spatially depend on entity elements such as administration region, community and street separately. Specially, these entity elements are hierarchical in spatial aspect. Similarly, economic elements, such as government, company and human, spatially depend on entity elements like building.
(2) Time dependencies. Entity element and economic element are not static but have time effect. The appearance of the time effect is expressed by event elements. For example, as shown in Fig. 1, the construction information, real-estate information

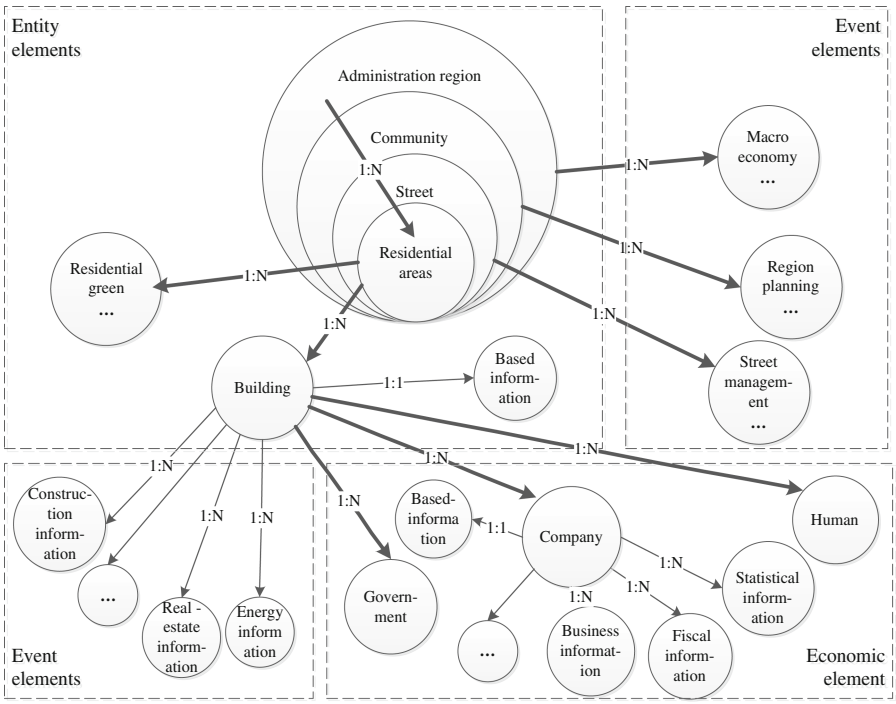

Fig. 1. The dependencies of urban elements

and energy information are events happened on the entity element of building along the time axis. The business information, fiscal information and statistical information are events happened on the economic element of company along the time axis.

(3) Dependencies among objects of these elements along sptio-temporal scales. An entity object consists of three basic characteristics at least: location, property and time status [14]. In another word, different temporal and spatial statuses map to different triples containing entity object, economic object and event object. The dependency can be expressed as follows.

$$O_{event} = (O_{time}, O_{location}, < O_{economy} > , O_{entity}) \tag{1}$$

where O_x denotes the corresponding object of type x. $O_{economy}$ is an optional item. For example, "planning results = (planning stage, level-2 area location, planning bureau, the north zone)" is a complete description of an event. In fact, all artificial entity objects in urban have similar experience timeline as "planning stage - design stage - implementation stage – completed stage - destruction stage". Each stage is an event node with spatial and temporal limitations. The target entity described by event nodes change from rough to fine in spatial scale with time flowing. In the planning stage, we

only require position accuracy to level-2 zone. But in the completed stage, we require position accuracy to level-6 with smaller size as a single building. Thus, the same location might refer to different entity objects due to different sptio-temporal scales.

3 The Integrated Code Construction of Urban Entity Elements

From Sect. 2, it could be inferred that event elements related to space are the interaction results between economic elements and entity elements, so each entity or economic object should own a unique identification to prevent redundancy and ambiguity when participating in city events. In smart city applications, an economic object has had an ID card or a corporate code as identity authentication. However, an entity object has multiple IDs in distinct systems, which brings various problems in data exchange and sharing process. Therefore, each entity object requires an ID like economic object to identify its uniqueness in different application scenarios.

An entity object can be treated as a geographic individual. In object-oriented theory, an object is a conceptual body with a unique identification. Each geographical sptio-temporal object compresses time, spatial, type properties and related behavior operation into a single individual. From that perspective, the urban entity code can be seen as a simple geographic object model composed of type, spatial and time codes as shown in Fig. 2.

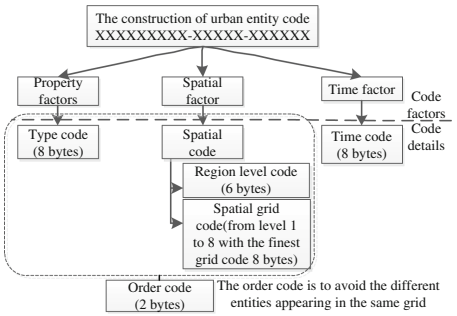

Fig. 2. The integrated coding framework of an urban entity object

From Fig. 2, it can be seen that type code, spatial code and time code correspond to property, space and time factors of an entity object respectively. Among them, type code and time code have consistent length once the initial works are executed before geocoding integration. But it's not the same as the spatial code. Spatial code consists of 2 parts: region level code and spatial grid code. The latter are a series of codes form level 1 to level 8 with the finest one 8 bytes. Its length varies with the spatial range of the object. To avoid different objects appearing in the same grid, the order code is necessary. More information will be presented in the next state.

4 Urban Entity Coding Algorithm

4.1 The Framework of Coding Integration

The UPPGP is a host platform for spatial data. When new vector data are entering, they are merged into the code database of the platform quickly. The vector data waiting for integrating should meet two conditions at least: (1) the data must have type attributes; (2) the geometric type (point, line or polygon) of the input data should map to the code dataset sharing the same type with the input data. The whole process contains three steps, followed by an illustration in Fig. 3(a). Any vector data can be merged into the data center by this method without changing the construct of the original data, so it can be easily used.

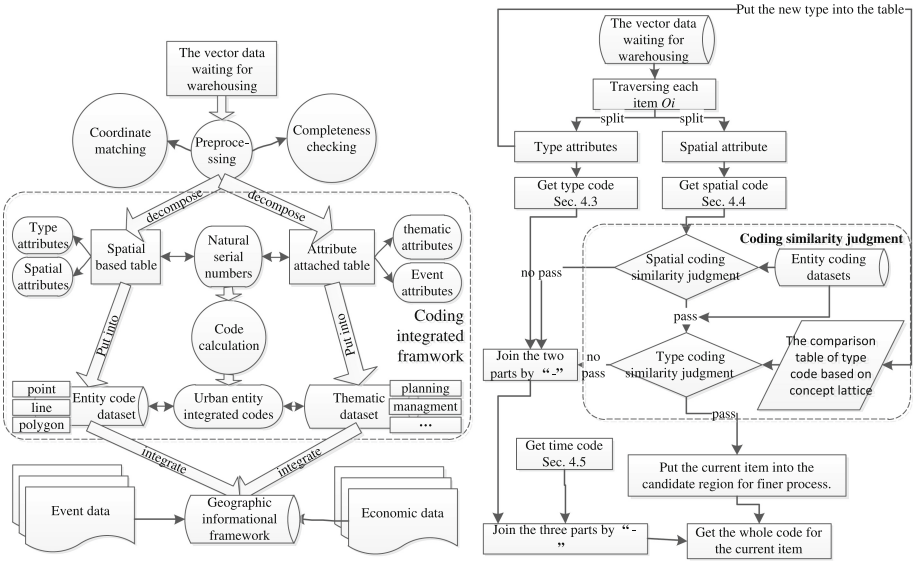

Fig. 3. The framework of coding integration

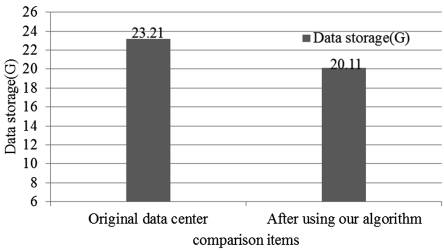

Fig. 4. Data storage improvement

First, pre-process the vector data. It is the level-1 integration proposed in Sect. 1. Rather than regurgitating, pre-process such as coordinate matching and data completeness checking before deeper coding process are discussed a lot in previous researches, so we don't present too much detail about that. **Second, decompose the data.** The process decomposes the data artificially into double parts: spatial based table and attribute attached table. The former only contains attributes that relate to the type and spatial issues of the data. The latter contains other attributes referring to the event and thematic issues of the data. The two parts are connected by the natural serial numbers or the numbers from their previous systems. **Third, calculate the integrated code and put the data into storage.** The vector data center of the UPPGP contains two parts: entity code dataset and thematic dataset. The former contains non-redundant vector data of point, line and polygon with unique codes. The latter contains 2-dimensional tables that related with different thematic domains such as planning and management. The two dataset are separated logically in data center. We calculate the code number according to the spatial based table and put the non-redundant data items into the entity code dataset. Besides, we put the attribute attached table into the thematic dataset as well. Then the two parts are connected by urban entity integrated codes presented in our paper. The details are shown in Fig. 3(b).

Step 1: decompose the type attributes into several properties to generate type code of the entity. In addition, find its semantic location in the concept lattice of urban entity elements. The flow is mentioned in Sect. 4.2.

Step 2: generate the spatial code by the spatial attribute according to the method mentioned in Sect. 4.3.

Step 3: do a similarity judgment by comparing the type code and spatial code of the current object with codes in the UPPGP database to determine whether it is a new object or a suspected existing one. The details are carried out in Sect. 4.4.

Step 4: to the new entity, generate time code by using the 8-byte storage time stamp, which is accurate to date. Then connect it with spatial code and type code by "-" and insert it into database. To the suspected existing object, move it to "the candidate region of duplicated objects (means the current object might be redundant)" to do finer processing.

The previous steps implement a rough filtering on the data content. To objects in the candidate region, level-3 integration can be implemented according to the detailed outline and other related attributes of the objects. From that point, the spatio-temporal geocoding method proposed in this paper is a transition level from level-2 to level-3 integration to make the computing complexity of the top integration much lower than before.

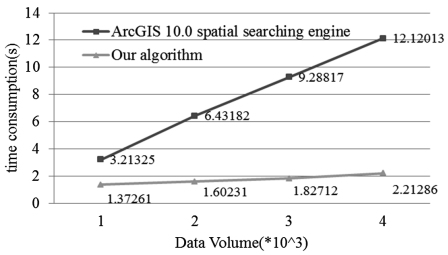

Fig. 5. Regional search improvement (street level)

4.2 Type Coding Algorithm Based on Concept Lattice

Type code expresses the essential attribute of the entity elements. In this section, we design a method to calculate the type code of urban entity element by building a property tree based on concept lattice and geographic ontology. This coding method can quantize the difference of massive entity elements. The processes are as follows.

First, collect the nature descriptions of urban geographical entity elements from the current standards to build the basic geographic concept dataset. The dataset will be enriched when new vector data are put into storage, since the type descriptions of the new data waiting for warehousing are also the source of the concept dataset.

Second, change the nature descriptions of each concept term into formal expressions. Each entity element has attributes that relate to materiality, reason, form, position, timeliness, and function [15]. Based on the attribute hierarchy tree, we denote each type as a list of formal words.

Third, Change each formal expression into binary code. If the current entity has the same attribute, the corresponding position is marked with "1"; otherwise it is marked with "0". It can be inferred that the granularity of the attributes effect the length of the type code. The longer the code is, the more certain the entity is. To avoid excess manual work, the numeric attributes like ">60 g" are ignored. But it is better to refine the attributes to improve the precision of similarity judgment. In addition, the granularity of type code is limited at the beginning usually.

Fourth, convert the binary code to 8-byte 64 hexadecimal code, with a cover of zeros in front if the code size is less than 8. The step can enlarge the capacity of the type code dataset in a limited storage circumstance.

At last, built the concept lattice for all entity type code and get the type code comparison table. The classification based on concept lattice [9] is used to the construct the code comparison table.

4.3 Spatial Coding Algorithm Based on Geocoding

Spatial code expresses the spatial characteristic of an entity object. It's an important part of the complete coding system. Current spatial coding methods include two types: regular grid geocoding and irregular grid geocoding. The former is based on the theory that the position of the point of interests (POIs) can be replaced by tiny areas. The method cuts the earth surface into regular and adjacent grids according to certain mathematical rules, and then gives each grid a unique code to represent the position of POIs in the grid. The "National Area Code (NAC)" developed by Canadian NAC company and GeoHash code proposed by Gustava Niemeyer both belong to this category [16]. The latter is represented by address geocoding method. It implements a conversion from descriptive geographic language (such as street or postal number) to spatial coordinates by building the coordinate correspondences between city administration units (like administrative area, street and house number) and their geometric center. The "Topologically Integrated Geographic Encoding and Referencing" [17] and "Postal Code Address Data" [18] both belong to that. The paper absorbs advantages of the two approaches. The processes are as follows.

First, calculate the Minimal Body Rectangular (MBR) for each entity object based on the spatial attribute. Second, to the scope larger than the street level (such as province or county level), use 6-byte postal code. To the scope smaller than the street level(such as residential area), consider not only postal codes but also regional Geo-Hash code [19] to cover tiny entities that are important but cannot be expressed by irregular grids, such as dustbins. Third, get the whole GeoHash codes arranged from level 1 to 8 combined with "-".

4.4 Coding Similarity Judgment

The coding similarity judgment is to distinguish the suspected redundant data items form the non-redundant ones. It contains two aspects:

(1) Spatial coding similarity judgment

From Sect. 4.3, we know that spatial code contains different size of indexed grids. It's a good idea to fully use the organized grid codes to get the similarity between two entity objects by calculating grid overlapping area in the same postal unit. The formula is:

$$Sim\_S(O_i, O_j) = \sum_{m=1}^{level} \frac{1}{2m}\left(\frac{GridSum(O_{im} \cap O_{jm})}{GridSum(O_{im})} + \frac{GridSum(O_{im} \cap O_{jm})}{GridSum(O_{jm})}\right) \quad (2)$$

where $Sim\_S(O_i, O_j)$ is the overlapping degree between the current object O_i waiting for warehousing and the target object O_j in the code dataset. *level* is the grid level. $GridSum(O_{im})$ is the grid sum of O_i in the current *level m*, $GridSum(O_{im} \cap O_{jm})$ is the cardinal number of the intersection of O_i and O_j in the current *level m*. If $Sim\_S(O_i, O_j) > \gamma$ (γ is the threshold), then the process goes to the next "type coding similarity judgment" step. Otherwise, the method combines the spatial code of the current entity with its type code and time code, and inserts the coded entity into database.

(2) Type coding similarity judgment

To entities passing the spatial coding similarity judgment, the coding method uses Eq. (3) [9] to calculate the type code similarity between the current object O_i and the target object O_j in the code dataset.

$$Sim\_T(O_i, O_j) = \omega_1 \left| \frac{O_i^{\triangleright} \cap O_j^{\triangleright}}{O_i^{\triangleright} \cup O_j^{\triangleright}} \right| + \frac{\omega_2}{DisMin(O_i, O_j)} \quad (3)$$

where $Sim\_T(O_i, O_j)$ is the type similarity between the current object O_i and the target object O_j. $O_i^{\triangleright}$ is the attribute set of O_i. $O_i^{\triangleright} \cup O_j^{\triangleright}$ is the intersection of O_i and O_j attribute set. $O_i^{\triangleright} \cup O_j^{\triangleright}$ is the union of O_i and O_j attribute set. $DisMin(O_i, O_j)$ is the minimum sum of edges from O_i to O_j in the attribute concept lattice. ω_1 and ω_2 are two adjustable parameters to meet the constraint of $\omega_1 + \omega_2 = 1$. If $Sim\_T(O_i, O_j) > \eta$ (η is the threshold), the current entity pass the judgment.

5 Applications and Experiments

The coding method has been successfully used in the UPPGP of Sino-Singapore Tianjin Eco-city already. It brings remarkable improvements to the data storage and searching efficiency for the UPPGP. The data storage improvement is shown in Fig. 4. In our platform, the volume of the data center is 23.21 G. After using our coding method, the data volume changes to 20.11 G. The Algorithm brings a 3.1 G storage decline for the vector data center. The searching improvement is shown in Fig. 5. Taking the block scale for example, we selected 4 cases of data volume with 10^3 intervals to test the efficiency of our method. The result shows that our method can save a lot of time compared with spatial searching engine of ArcGIS 10.0. We get similar results in other spatial scales as well. Furthermore, Multi-level searching based on this code is shown in Fig. 6.

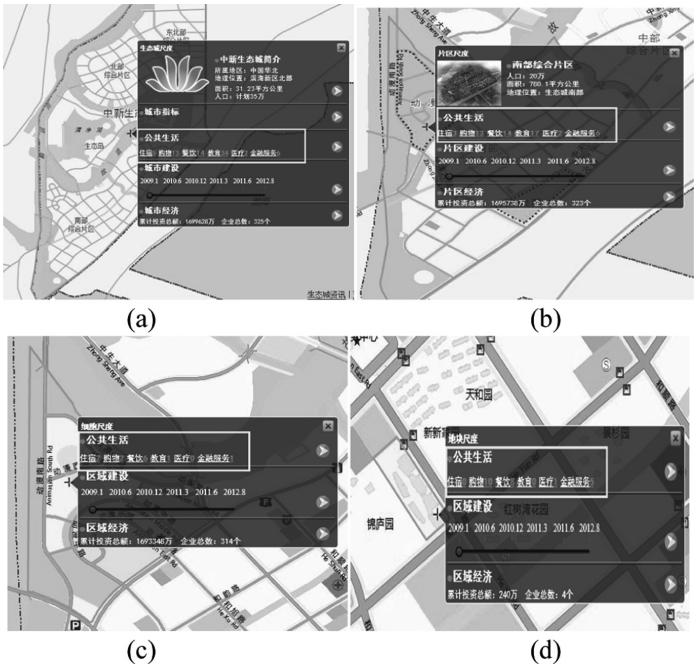

Fig. 6. Multi-level searching based on our code. (a) shows the function window on the global scale. It can be seen that 4 types of indicators related on the whole regional scale: urban comprehensive indicators, urban entity statistic indicators, urban construction indicators and macro-economic indicators. Similarly, (b) shows the function window referred to indicators on the district scale; (c) shows indicators on the block scale, and (d) shows indicators on the plot scale. Each spatial scale corresponds to different function windows.

Firstly, each spatial entity object is implemented the two-step judgment before storage. If it does not pass the test, the object will be given a unique code and inserted into the database. Otherwise, the duplicates will be put into the candidate region for

finer process. By that way, the vector data are cleaned. Even though the method is taken as a rough filtration, it is really an effective way to avoid redundancy.

Secondly, we get the information about type, location and generated time of the target object from the code easily, and then do a rapid query based on the same part of the current entity code to find related ones. The matching and query tasks are similar to the coding process. However, the application case is only the tip of an iceberg.

6 Conclusion

From Sect. 5, it can be seen that the spatio-temporal geocoding method is an effective way to solve data redundancy and inefficient retrieval problems during the data integration. This method is based on the content and inner connections among urban elements. We ensure the key role of urban entity elements in connecting event elements and economic elements, and abstract out a geographic entity model composed of type code, spatial code and time code. In our Algorithm, type code is calculated by decomposing entities' attributes formally and using concept lattice to generalize the semantic position of the entity, space code is calculated by regional GeoHash code to generalize the space range of the object, and time code is calculated by recording the storage time stamp of the object. Meanwhile, we also consider the order code to avoid different entities appearing in the same grid. When doing geocoding integration, the method decides whether the outer data items exist in the original code dataset or not. To non-existing records, our method inserts them into database according to the coding rules.

The method doesn't change the physical structure of the original data, but implements loose integration by the unique identification designed in the paper. To emphasize, the method cannot replace the top level integration completely, but can be regarded as a transition from the original level-2 to level-3 integration methods. Due to this geocoding process, the UPPGP can execute the semantic and geographic integration more easily. That is also our next topic, which will be discussed more in future studies.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 2011FU125Z24.

References

1. Uitermark, H.T.: Ontology-based geographic data set integration. Ph.D. Dissertation, Deventer, The Netherlands (2001)
2. Schönberger, V.M., Cukier, K.: Big Data: a Revolution That Will Transform How We Live, Work and Think. John Murray, England (2013)
3. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans. Knowl. Data Eng. **9**(24), 1537–1555 (2012)
4. Grayson, T.H.: Address Matching and Geocoding. Massachusetts Institute of Technology Department of Urban Studies and Planning, vol. 14 (2000)

5. Huh, Y., Yang, S., Ga, C., Yu, K., Shi, W.: Line segment confidence region-based string matching method for map conflation. ISPRS J. Photogrammetry Remote Sens. **78**, 69–84 (2013)
6. Kyung, M.J., Yom, J.H., Kim, S.Y.: Spatial data warehouse design and spatial OLAP implementation for decision making of geospatial data update. KSCE J. Civil Eng. **16**(6), 1023–1031 (2012)
7. Ok, G.H., Lee, D.W., You, B.S., Bae, H.Y.: A spatial data cubes with concept hierarchy on spatial data warehouse. Korea Inf. Process. Soc. **1**(13), 35–38 (2006)
8. Egenhofer, M., Mark, D.: Naive geography. In: Kuhn, W., Frank, A.U. (eds.) COSIT 1995. LNCS, vol. 988, pp. 1–15. Springer, Heidelberg (1995)
9. Kang, X.P., Li, D.Y., Wang, S.G.: Research on domain ontology in different granulations based on concept lattice. Knowl. Based Syst. **27**, 152–161 (2012)
10. Kokla, M., Kavouras, M.: Fusion of top-level and geopraphical domain ontologies based on context formation and complementarity. Int. J. Geogr. Inf. Sci. **15**(7), 679–687 (2001)
11. Filin, S., Doytsher, Y.: A linear mapping approach to map conflation: matching of polylines. Surveying Land Inf. Syst. **59**(2), 107–114 (1999)
12. Frank, A.U.: Qualitative spatial reasoning about distances and directions in geographic space. J. Vis. Lang. Comput. **3**(4), 343–371 (1992)
13. Shao, J., Yang, L.-N., Peng, L., Yao, X.-J., Zhao, X.-L.: Research of data resource management platform in smart city. In: Bian, F., Xie, Y. (eds.) GRMSE 2014. CCIS, vol. 482, pp. 14–22. Springer, Heidelberg (2015)
14. Butenuth, M., Gosseln, G., Tiedge, M., Heipke, C., Lipeck, U., Sester, M.: Integration of heterogeneous geospatial data in a federated database. ISPRS J. Photogrammetry Remote Sens. **62**(5), 328–346 (2007)
15. Wang, H., Li, L., Zhu, H.H.: The Key Research of National Fundamental Geographic Information Ontology (in Chinese). Science Press, Beijing (2011)
16. Parker, N.: A Look at NAC Geographic Directions Magazine, US (2004)
17. Census Bureau TIGER. http://www.census.gov/geo/maps-data/data/pdfs/tiger/tgrshp2013/TGRSHP2013_TechDoc.pdf
18. Canada Postal Guide-Addressing Guidelines. http://www.canadapost.ca/tools/pg/manual/PGaddress-e.pdf
19. Jing, A., Cheng, C.Q., Song, S.H., Chen, B.: Regional query of area data based on Geohash (in Chinese). Geogr. Geoinf. Sci. **29**(5), 31–35 (2013)