The Journal of GIS Association of Korea, Vol. 13, No. 4, pp. 355-364, December 2005

A GIS Vector Data Compression Method Considering Dynamic Updates

Woo-Je Chun* · Yong-Jin Joo** · Kyung-Ky Moon*** · Yong-Ik Lee**** · Soo-Hong Park*****

ABSTRACT

Vector data sets (e.g. maps) are currently major sources of displaying, querying, and identifying locations of spatial features in a variety of applications. Especially in mobile environment, the needs for using spatial data is increasing, and the relative large size of vector maps need to be smaller. Recently, there have been several studies about vector map compression. There was clustering-based compression method with novel encoding/decoding scheme. However, precedent studies did not consider that spatial data have to be updated periodically. This paper explores the problem of existing clustering-based compression method. We propose an adaptive approximation method that is capable of handling data updates as well as reducing error levels. Experimental evaluation showed that when an updated event occurred the proposed adaptive approximation method showed enhanced positional accuracy compared with simple cluster based compression method.

Keywords : Vector compression, clustering, GIS, mobile, CBC, dictionary-based

요 약

모바일 기기의 제한적 환경에서 공간데이터의 활용을 극대화하기 위해 벡터데이터의 압축에 대한 연구가 최근 이뤄지고 있다. 이 중 군집화 방법을 이용한 벡터데이터 압축은 기존 압축방법과 다른 새로운 형태로 주목을 받고 있다. 그러나 현재까지 연구는 데이터의 동적인 갱신이 고려되지 않았다. 본 연구는 기존의 군집화 방법을 이

* Master course, Department of Geoinformatic Engineering, Inha Univ. (E-mail : woojechun@hotmail.com)
** Ph. D. Candidate, Department of Geoinformatic Engineering, Inha Univ. (E-mail : ccmdrum@netian.com)
*** Master course, Department of Geoinformatic Engineering, Inha Univ. (E-mail : starmkk@naver.com)
**** Master course, Department of Geoinformatic Engineering, Inha Univ. (E-mail : a78leekey@naver.com)
***** Professor, Department of Geoinformatic Engineering, Inha University (E-mail : shpark@inha.ac.kr)

용한 벡터데이터 압축방법의 문제점을 파악하고, 데이터의 동적인 갱신이 고려된 압축 방법을 제시하였다. 실험을 통한 결과는 갱신이 발생하였을 경우 제안된 방법이 더 좋은 결과를 나타냄을 확인할 수 있었다.

주요어 : 벡터데이터, 지리정보시스템, 압축, 군집화, 사전기반

1. Introduction

The confluence of ever smaller and more powerful mobile computing devices equipped with broad band wireless communication connectivity are leading the world into a ubiquitous computing era[9]. Improved hardware and networking of mobile devices make Location-based Services(LBS), Car Navigation Systems(CNS), and Telematics Systems appeal to customers nowadays. Spatial data management and processing in such applications require simpler and more compact data models[16].

Adoption of vector data in mobile environments is increasing due to the relatively smaller size of data volume and better representation for displays[16]. Even though vector data are compact and small, storage limitations of current mobile devices still exist[16]. A current map database of CNSs or telematics devices need several hundred mega bytes of storage space as minimal and tend to increase the storage requirement radically. To overcome such storage limitations, desirable data compression techniques are needed.

Some of previous studies introduced vector data compression methods, in which clustering-based compression (CBC) methods[16] adopting the lossy compression framework are appropriate and applicable to many spatial applications. Existing CBC methods provide a good compression ratio, a reasonable accuracy level and a simple and fast decoding process. Another clustering-based method[10], called modified CBC, adopting scalar clustering framework based on CBC provides a better accuracy level with enough number of cluster. However, the resulted error level of both methods may not be suitable for some applications (e.g. car navigation). More critically existing methods have a severe limitation for incremental data additions and dynamic data updates.

This paper explores the problems of existing modified clustering-based compression (CBC) method in the context of the mobile spatial application environment. We propose an adaptive approximation method that is capable of handling data updates as well as reducing the errors levels.

2. Related Work and Problem Definition

2.1 Cluster-based Compression Approach

Shekhar et. al[16] explored issues and problems of existing vector map compression methods

and developed a new clustering based compression technique. This technique adopted the dictionary based on the FHM (Fibonacci, Huffman, and Markov) curve compression algorithm. The entries of dictionary were produced using the K-means clustering algorithm, which represent the centroids of clusters calculated from the coordinates of differential vectors for each line segments.

Their CBC method provides a fast decoding scheme that simply looks up entries of dictionaries and restores original coordinates of vertices with already calculated centroids except the starting point. Experimental results showed that their lossy compression method provides a reasonable error level for a small scale linear data set.

2.2 Modified CBC Method

Lee[10] explored issues and problems of existing CBC methods and took a new approach. The overall framework of the modified CBC method was maintained for this approach except for the clustering target. Instead of using the differential vectors of each line segment suggested in the previous CBC method, it computed two scalars (i.e., lengths and angles) from the differential vectors and applied clustering algorithms into them respectively. Clustering angles and lengths have two benefits. The first, it diminishes the range of scalar values, especially in the case of angles. It can make clusters more compact so that resulting encoding errors can be reduced. Secondly, entries of two separated dictionaries can extend the possible number of encoding/ decoding values with combining the two keeping the same number of entries in total. These benefits enabled this method to have more accurate result.

2.3 Reference Line Approach

Akimov and et. al[1] devised an enhanced CBC algorithm by introducing the concept of reference lines based the polygonal approximation method. They first produced a series of references lines consisted of only two end points and used them for as a basis of the encoding scheme. The reference lines can be considered as a lower resolution representation of the original data and will be stored in the compressed file. Information about the reference lines were used to extract the differential vectors of the original vertices.

Further, a dictionary construction is performed in a step wise manner based on each reference lines to cluster the differential vectors using a dynamic programming algorithm. They showed that the resulted encoding errors were greatly reduced compared to the CBC method slightly sacrificing the compression rates.

2.4 Problem Definitions

The previous works fundamentally assume that the operational environments of these methods should be static. Additional data insertion and updates for the original data are

not considered. If some new data are selectively added and updated in those methods, the whole compression procedures should be performed again, which may not be possible in a mobile environment. Otherwise we could use existing entries of the dictionary. However, the existing clusters may not reflect the differential vectors of the new data and produce large errors consequently.

In addition to the problem of considering the dynamic data characteristics of mo-bile environments, existing compression methods need to be enhanced the accuracy of the compressed data that enables us to apply them into the applications requiring relatively large scale vector data sets such as car navigation or telematics services. Current error levels of those compression methods may be reduced with some sacrifice of the compression rate.

We found that previous methods have an elegance framework applicable to many mobile spatial applications. However, considering the necessity of adding or updating data in the mobile applications, an extended approach that does not require re-clustering the whole data and still supports an efficient encoding/decoding scheme should be explored. This approach will mainly focused on how to reflect newly input data into the dictionary adaptively without re-clustering and changing the entries of dictionary by way of incorporating additional approximation components.

3. An Adaptive Approximation Approach

3.1 Methodology Framework

The adaptive approximation approach we propose here is based on the previous modified CBC method. The overall framework of the modified CBC method in which extracting differential vectors and scalars, grouping the scalars with K-means cluster algorithm, and a dictionary based encoding/decoding scheme were maintained for this approximation approach. Besides the basic components, approximation functions and factors were incorporated to enable us both to use existing clusters and minimize errors induced from newly input data.

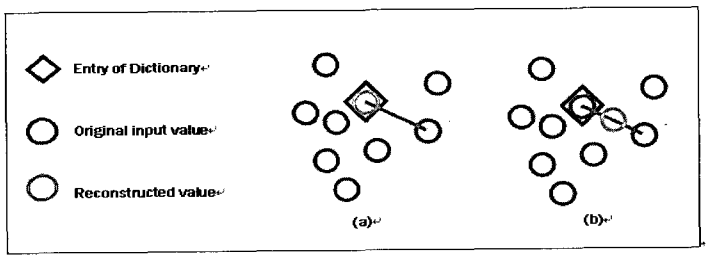


Figure 1. (a) modified CBC method, (b) adaptive approximation method

3.2 Adaptive Approximation Functions and Factors

Approximation functions and factors are added to minimize the differences between input values and their closest clusters. The major role of an approximation factor in this new approach is to record the closeness between input values and their representing clusters. It enables us to reconstruct the values which are closer to the original input values than entries of dictionary when decompression is processed. In other words, it enables us to avoid adhering to entries of the dictionary which have large errors. The brief concept of adaptive approximation is shown in figure 1.

In figure 1, reconstructed value in (b) is computed with an approximation factor, an approximation function, and the entry of dictionary. You can see that reconstructed value in (b) is more faithful to the original value than one in (a). In consequence, we can have two advantages while sacrificing compression rate.

The first, we can enhance accuracy level. In modified CBC method, several values, which are not same but are included in one common cluster, are reconstructed as one value, such as a centroid value of the cluster. When the number of cluster is not enough, this deteriorates accuracy level dramatically in modified CBC method. However, in adaptive approximation method, we can have respective reconstructed values for the each input values in the same cluster. It can overcome shortage of the number of cluster.

Secondly, we can cope with data updates efficiently. In modified CBC method, when data update occurs, either whole compression procedures should be performed again, or existing entries of the dictionary should be

```
Designed Algorithm
1)  For each segment do
2)    Separate base point and sequence of delta values
3)  end;
4)  For each delta values do
5)    Separate angle and length from delta values
6)  end;
7)  Do
8)    K-means clustering on set of angle, length
9)    each mean is a dictionary entry
10)   for each original angle, length do
11)     encode angle, length using dictionary
12)   end;
13)   for each original angle, length do
14)     find approximation value using dictionary and encoded angle,length
15)     encode approximation arrays for angle, length
16)   end;
17) end;
```

Figure 2. Compression algorithm of the adaptive approximation method

used to represent updated data. The first one may not be possible in a mobile environment because of expensive computational cost, and the second one may not reflect the new data and produce large errors. However, in adaptive approximation method, without re-clustering, we can use existing clusters to incorporate new data with the help of approximation factors. When additional data insertion or updates occur, we store not only the closest cluster information of existing dictionary but also approximation factors for the data. Even though it sacrifices compression rate, it avoids both huge computational cost by re-clustering and large errors by inappropriate cluster.

The entire algorithm is shown in figure 2.

The closeness between a scalar value and the closest entry of dictionary is calculated by

$$\mu(x) = \frac{1}{S} \times \frac{x}{D}$$

where S is the scale of scalar, x is the input value, D is the entry of the dictionary. In this study, we assumed that the largest scalar of the updated data was smaller than twice of the maximum value of the existing dictionary, so we set S as 2. The resulted μ (x) is in range between 0 and 1. A real number is cumbersome for us to encode because of storage space. We need to change it lighter. To solve this problem, we used the approximation function which is calculated by

$$f(t) = \frac{t}{n}$$

where t is an approximation factor and n is the total number of approximation factors. Both t and n are integer numbers. When we find the closest f(t) to μ(x), we encode t as the approximation factor. In decode process, the decoded scalar is calculated by

$$R = D \times S \times f(t)$$

where R is the decoded scalar value, D is the entry of the dictionary, S is the scale, f(t) is the result of the approximation function, t is the approximation factor.

4. Experiments and Results

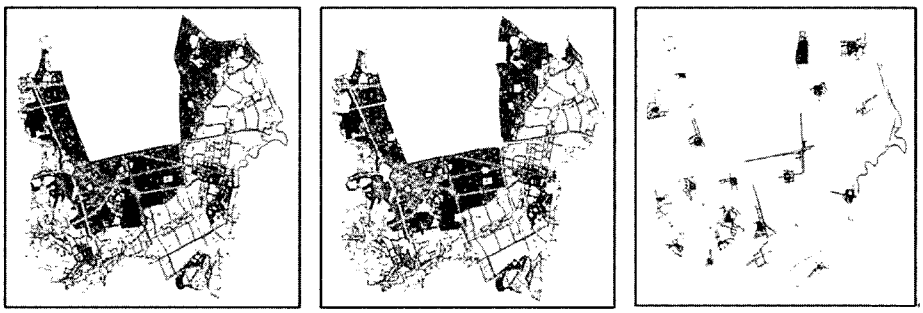
4.1 Experiment Data Sets

As we focused on data updates, we carefully selected two datasets which had distinct characteristics. The first dataset has house and building outlines in part of Seoul which are from 1:1,000 digital topographic maps. It includes 72,016 polygons and 566,607 individual coordinate points. In this map, the extracted scalars (lengths and angles) have narrow distribution. This may cause that the existing dictionary covers updated data without significant errors.

The second dataset is land parcel boundaries of the part of Seoul area. It includes 25,914 polygons and 228,722 individual coordinate points. This dataset includes road edges which have wide distribution in length. Therefore, the

(a) (b) (c)

Figure 3. Dataset #1: House & Building Outlines (regular polygons) - (a) original data, (b) modified data, (c) added data



(a) (b) (c)

Figure 4. Dataset #2: Land Parcel Boundaries (irregular polygons) - (a) original data, (b) modified data (c) added data

existing dictionary may not cover updated data.

To setup update scenario, we set aside 10% of the original data and used it as an added dataset. Each experiment dataset is shown in figure 3, 4.

The major difference between our technique and the modified CBC method is that we use approximation factors. Therefore, modified CBC method was compared to the proposed adaptive approximation method in order to check the influence of the approximation factor.

4.2 Experimental Results

Even though modified CBC method has advantages in a variety of the delta values and better RMSE, it is still weak to reflect added data. If the length of updated data is larger than the maximum value of clustered length, the difference between two scalars becomes error. There is no way to reduce it in modified CBC method. To overcome this problem, the proposed method uses a scale

factor. With an update scenario, we tested modified CBC method and the proposed method.

Table 1 and 2 show the results of modified CBC method. RMSE was reduced as entry grew. However, considering maximum error, we can find no improvement in table 2. The reason is that the existing dictionary of dataset #2 does not cover updated data while one of dataset #1 covers. Figure 5 shows the region of the maximum error.

Table 1. The result of modified CBC method on dataset #1

Entry		Compression rate (%)	RMSE (m)	Maximum error (m)
128	128	79.09	0.3299	6.6606
256	256	77.88	0.1548	2.3161
512	512	76.62	0.0724	1.7548
1024	1024	75.28	0.0342	1.5384

Table 2. The result of modified CBC method on dataset #2

Entry		Compression rate (%)	RMSE (m)	Maximum error (m)
128	128	80.01	1.2131	492.9530
256	256	78.78	0.5355	489.0379
512	512	77.35	0.2480	489.4244
1024	1024	75.69	0.1182	489.7993

Table 3 shows the result of proposed method. It reduces the maximum error as entity and total approximation factors grow.

Figure 6 shows the maximum errors of each method. The graph shows that the maximum error is only reduced by the approximation factor and that total entry of the dictionary does not make any influence. In figure 7, at the same entry, more approximation factors reduce RMSE. It means approximation factor can improve accuracy level of CBC method.

Table 3. The result of proposed method on dataset #2

Entry	Total approx. factors	Compression rate (%)	RMSE (m)	Max Error (m)
128	16	75.27	0.9214	47.7828
	64	72.85	0.4612	12.8082
	256	70.43	0.1460	8.4512
256	16	73.94	0.4786	49.1842
	64	71.53	0.3116	9.3713
	256	69.11	0.1497	7.9913
512	16	72.51	0.2323	48.7672
	64	70.10	0.1914	7.9816
	256	67.68	0.1122	6.6930
1024	16	70.86	0.1058	48.5483
	64	68.44	0.0904	7.6971
	256	66.02	0.0709	6.6930

Figure 5. Land Parcel Boundaries: left image is the original data, and right image is the reconstructed data by modified CBC method

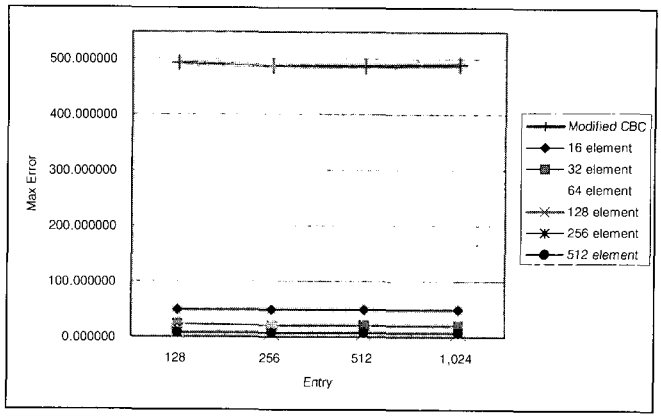


Figure 6. Maximum error on Land Parcel Boundaries

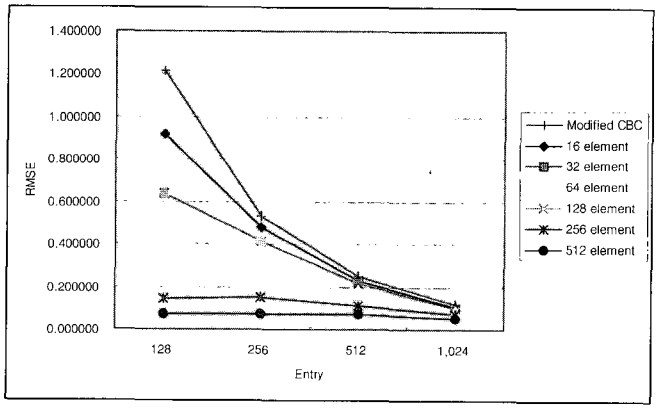


Figure 7. RMSE on Land Parcel Boundaries

5. Conclusions

In this paper, we have explored the problem of modified CBC method for the data update. Then we have proposed the use of approximation factor and scalars. This proposed method showed the reflection mechanism of the updated data, and made the data more reliable by eliminating unpredicted huge errors. We predict to use this method for the area where frequent renovations occur.

In future work, we would like to experiment on various datasets. As modified CBC method should fix the cluster number beforehand, adaptive approximation method should fix the parameters such as the total number of approximation factors and the scale factor. Therefore, after experiments on various data, which have different subject and different size, we can determine objective and proper parameters.

References

[1] Akimov, A., Kolesnikov A., Franti P., 2004, "Reference Line Approach for Vector Data Compression", Proc. Int. Conf. Image Processing-ICIP'04, Singapore, October 2004. pp. 1891-1894

[2] Clarke, K. C., 1990, "Analytical and computer Cartography," Practice-Hall

[3] Cox E., 1995, "Fuzzy Systems Handbook"

[4] Douglas, D. H., Peucker, T. K., 1973, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," Canadian Cartographer, vol. 10, pp. 110-122

[5] Freeman, H., 1961, "On the Encoding of Arbitrary Geometric Configurations," IRE Trans. Electronic Computers, Vol. EC-10, pp. 260-268

[6] Hair, J. F., Anderson, R. E., Thatham, R. L., and Black, W. C., 1992, "Multivariate Data Analysis (3rd ed.), New York: Macmillan Publishing Co.

[7] Jain, A. K., Murty, M. N. and Flynn PJ., 1999, "Data Clustering: A Review", ACM Computing Surveys, Volume 31, Issue 3, pp. 264 - 323

[8] Jan, J., Kanber, M., 2000, "Data mining concepts and Techniques", Morgan Kaufmann

[9] Kim, M.R., Choi, J.O., 2002, "Database: Design and Implementation of Client/Server System for Mobile Vector Map Services", Korea Information Science Society Vol. 9, pp. 819-826

[10] Lee, D.H., 2005, "Vector Data Compression Using a Clustering Method", Inha University Geo-Informatic Engineering

[11] Lee, J., Wong, D. W. S., 2000, "Statistical analysis with ArcView GIS", John Wiley & Sons Inc.

[12] Lu, C. C., Dunham, J. G., 1991, "Highly Efficient Coding Schemes for Contour Lines Based on Chain Code Representations," IEEE Transactions on Communications, 39(10): 1511-1514

[13] Macqueen, J., 1967, "Some methods for classification and analysis of multivariate observations.", In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 281 - 297

[14] OGIS, 1999, Open GIS Consortium: Open GIS simple features specification for SQL (Revision 1.1)

[15] Sayood, K., 2000, "Introduction to Data Compression", Morgan KaufMann Publishers

[16] Shekhar, S., Huang, Y., Djugash, J. and Zhou, C., 2002, "Vector Map Compression: A Clustering Approach", Proceedings of the tenth ACM international symposium on Advances in geographic information systems, pp. 74 - 80

[17] Weibel, R., 1987, "An adaptive methodology for automated relief generalization," Proceedings, AUTOCARTO 8, Eighth International Symposium on Computer-Assisted Cartography, Baltimore, MD, pp. 42-49