Vector Map Compression: A Clustering Approach *

Shashi Shekhar, Yan Huang, Judy Djugash, Changqing Zhou†
Department of Computer Science and Engineering
University of Minnesota
200 Union ST SE, 4-192
Minneapolis, MN 55414
[*shekhar*, *huangyan*, *djugash*, *czhou*]@cs.umn.edu

ABSTRACT

Vector maps (e.g. road maps) are widely used in a variety of applications such as Geographic Information Systems(GIS), Intelligent Transportation Systems(ITS) and mobile computing. However, the relatively large size of vector maps has in some cases negatively impacted their usage and application in these systems because of the small storage available with mobile wireless devices or the limited bandwidth of the data transportation. In these cases, data compression techniques need to be applied on these vector maps to handle larger datasets and faster data transportation. Among all the data compression techniques, dictionary-based compression is a good candidate since encoding and decoding do not need a significantly large amount of computing resources. This paper explores the problem of dictionary design for dictionary based vector map compression. We propose a novel clustering-based dictionary design which adapts the dictionary to a given dataset, yielding better approximation. Experimental evaluation shows that when the dictionary size is fixed, the proposed clustering-based technique achieves lower error compared with conventional dictionary compression approaches.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Data compaction and compression

General Terms

Algorithms design

*Supported in part by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory Cooperative agreement number DAAD19-01-2-0014. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred.

†Corresponding author(email: czhou@cs.umn.edu).

GIS'02, November 8–9, 2002, McLean, Virginia, USA.

Keywords

vector map compression, clustering, dictionary design

1. INTRODUCTION

Mobile computing devices, e.g. personal digital assistants (PDA) and in-car navigation units, require access to spatial datasets [5, 10, 11, 13, 14] such as vector maps for location-based services. An example of a query would be "Where is the nearest gas station?" Vector maps, e.g. road maps, consist of a collection of points (e.g. road intersections), line strings (e.g. center line of road segments connecting intersections), and polygons (e.g. boundaries of parks, cities, lakes etc). In contrast, raster maps use image or matrix representation. In mobile computing, vector maps are more desirable because of their smaller size. However, mobile devices normally have small storage and limited data transportation bandwidth. Vector maps are still too big for them in many cases. For example, a typical PDA such as Palm III and Palm V has 2 to 64 megabytes of storage. The size of a city map (e.g. in Microsoft pocket street format) is usually around 0.5 to 2 megabytes. A PDA can store only a few maps in the space remaining after the space occupied by the operating system and other essential software and data.

Compression techniques for vector maps can allow PDAs to carry larger subsets of vector maps or free up memory for other datasets, e.g. appointments, address books, etc. Compression techniques for vector maps can also reduce the communication cost of downloading new maps to the PDA, possibly over low-bandwidth wireless channels (e.g. beaming, cell phone modems).

The goal of vector map compression is a compact representation of map data, with possibly some limited sacrifice of spatial accuracy. Lossy compression schemes are acceptable since cartographers routinely use map simplification to highlight key features in a map by introducing bounded distortions, e.g. errors in the location of spatial objects. However, simple decoding schemes should be allowed due to the limited computational resources of popular mobile devices. This constraint makes dictionary-based compression techniques attractive as long as the error of approximation can be controlled.

This paper explores the problem of designing dictionaries for the dictionary-based compression of vector maps with minimal approximation of errors. It proposes the use of clustering techniques[2] (e.g. K-mean clustering [7]) to identify dictionary entries while minimizing errors of approximation for locations of spatial objects in the vector map. We formally show that this proposed dictionary construction approach often yields a lower error of approximation than the error from conventional fixed dictionary techniques. Experimental results with a road map representing major US highways confirm the superiority of the proposed method in yielding lower errors of approximations when absolute delta vectors are used to encode linestrings.

1.1 Related Work and Our Contributions

Based on the data format of the maps, map compression techniques can be divided into two groups, namely raster map compression and vector map compression. Raster map compression techniques manipulate a raster matrix to get a concise representation. Vector map compression includes techniques such as line simplification, chain coding, and dictionary based compression [12]. Line simplification [6, 15] and chain coding [16] are used with paper maps and other line-drawings containing curves. These techniques approximate curves by a sequence of straight lines. The straight lines are from a fixed collection, e.g. vertical, horizontal, diagonal. Line-simplification and chain-codes often eliminate original points and add new points as a side-effect. FHM (Fibonacci, Huffman, and Markov) is a chain-coding-based [8, 16] algorithm designed for signature compression. Given a signature, the FHM algorithm uses a dictionary of line segments with fixed slopes and lengths. It replaces each line segment or group of consecutive line segments in the signature by searching the dictionary for the best fitting one.

However, digital vector-maps usually do not contain curves; rather, they consist of points, line-strings and polygons. Adding or removing points may not be allowed in compression since the map error assessments [3] are based on a comparison of equivalent points. Despite this constraint, the dictionary approach used in the FHM [9] algorithm can still be used. Dictionary-based algorithms construct a dictionary to match dictionary entries to line segments. When a dataset is decompressed, the dictionary is searched to find the data associated with the dictionary indices. The FHM dictionary, as a static dictionary, does not consider data distribution (slopes and offsets of segments in the curves) but rather uses a collection of line segments organized by a set of squares. If the distribution of line segments in a dataset does not match well with the dictionary entries of FHM, the errors of approximation can be large. In such a scenario, an alternative method which could incorporate the data distribution in the dictionary design would achieve lower error than the FHM method.

In this paper, we propose a clustering-based map compression method which adapts a dictionary to a given dataset. This method yields better approximations of the original map data, leading to lower error. We provide an experiment design and evaluation on a real dataset. Experimental results show that for a fixed size dictionary, the clustering-based dictionary construction compression method achieved lower error than a static dictionary compression such as the one used by the FHM algorithm. The basic idea was presented in an extended abstract [12] and a detailed description is presented in this paper.

1.2 Outline and Scope

In section 2, we discuss issues related to vector map compression. Different dictionary building schemes are described in section 3 followed by the experiment evaluation and comparison in section 4. We conclude in section 5.

The concepts of chain coding and line simplification are beyond the scope of this paper, though if used might lead to better minimization of errors. Also, the scope of this paper is limited to data comprised of line segments. Datasets which include curves are not addressed here.

[2] For a summary of clustering algorithms please refer to [7]

2. VECTOR MAP COMPRESSION ISSUES AND PROBLEM DEFINITION

Typically, a vector map is composed of objects represented by points, lines, polygons, and graphs. There are texts associated with each object. In the map, we can observe the metric properties of the objects in the map such as distance as well as topological properties [17] such as crossing, closed, adjacent, and disjoint objects. A map also has a scale and usually supports operations such as zoom in or zoom out.

When a vector map is compressed, certain properties such as the topological properties need to be preserved as much as possible. Roads which intersect should still appear to intersect after compression and de-compression of the map. When we compress the metric part of a map, the topological part should be preserved as much as possible. Figure 1 shows a framework for obeying this constraint. Vector map data is separated into topological data (e.g. nodes, edges) and geometric data (e.g. sequence of shape points for an edge). Lossy compression techniques may be used for geometric data but not for topological data. We focus on geometric data from here onwards, as the size of geometric data is usually much larger than the size of topological data.

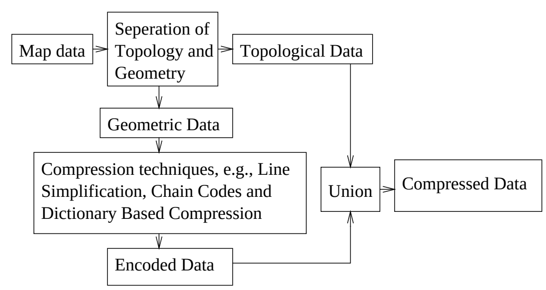

Figure 1: Map Compression Framework

Compression techniques for geometric data need to observe constraints as well. National Map Accuracy Standards require that maps produced by all federal agencies comply with published standards such as horizontal and vertical accuracy. For example, the horizontal accuracy standard requires that the positions of 90 percent of all points tested must be accurate within 1/30th of an inch on a map at greater than 1:20,000 scale and 1/50th of an inch on a map at 1:20,000 scale or smaller. Maps produced through compression and de-compression also need to maintain a level of user tolerant accuracy. These accuracy levels differ according to different application domains. Users of roads maps for travelling may have less stringent accuracy requirements than users of maps for road construction.

We define the problem of designing a dictionary for vector map compression as follows: Given a vector map and a specific dictionary size, we need to find a dictionary for the dictionary-based encoding scheme. The objective is to minimize the error of approximation for the given data. In achieving this objective, we need to keep in mind the constraints, namely that the topology of the map needs to be preserved. For example, road intersections on a road map need to be maintained. Similarly, if we have a map consisting of all the states in the United States represented as polygons, then these polygons need to remain closed.

3. DICTIONARY DESIGN TECHNIQUES

3.1 Framework

The overall framework for a dictionary-based compression technique is presented in Figure 2. Given a line string or polygon in an uncompressed map form, we first convert the coordinates in the uncompressed map into a base point followed by a sequence of differential vectors. The differential vectors may be determined by the vector differences between the current point and the previous point or the first point. We denote the former as $\Delta_{i,0}$ or Delta(i, 0) and the latter as $\Delta_{i,i-1}$ or Delta(i, i-1). Now we have a set of objects in the form of a BASE point followed by a series of differential vectors. The series of differential vectors produced can then be encoded to produce the compressed data by passing through a two-step process. First a dictionary of a given size needs to be constructed for the produced set of vectors, as shown in Figure 2. In the next step, the dictionary is used to encode the original dataset to produce the encoded, compressed data representation.

To illustrate, consider the road segment given in Figure 3. The dataset in uncompressed form can be represented as a series of coordinates of the form (5,5),(6.3,6),(7.6,5),(9.1,8.5). Converting this to a base point followed by a sequence of vector representations using the formula $\Delta X = X_i - X_{i-1}$ and $\Delta Y = Y_i - Y_{i-1}$, we get ((5,5),(1.3,1),(1.3,-1),(1.5,3.5)). Now leaving the base point as it is, the differential vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ are encoded based on the closest entries from the dictionary. This representation often compounds errors of approximation. For example, errors of approximation for the last point are higher than those for earlier points. A different representation uses differential vectors ($\Delta X_i = X_i - X_0, \Delta Y_i = Y_i - Y_0$), where (X_0, Y_0) is the first point. The road segment in Figure 3 can be represented as ((5,5),(1.3,1),(2.6,0),(4.1,3.5)). This representation allows direct control of errors of approximation during dictionary construction since the encoding of each differential vector is independent of the encoding of the other differential vectors. This compressed representation along with the chosen dictionary can be used later to decode the data for the purpose of analysis and error computation. In the following subsections, we present two approaches for dictionary construction.

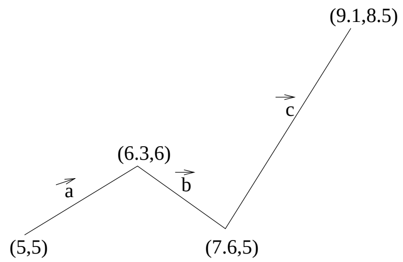

Figure 3: Example of a Road Segment

3.2 Dictionary Based on FHM Curve Compression

The FHM (Fibonacci, Huffman, and Markov) method based on multiring chain coding [9] is designed for compressing signatures. Squares of specific sizes are built around the current anchor point based on Fibonacci numbers. Thus a set of nested squares of sizes 1X1, 2X2, 3X3, 5X5, 8X8 and 13X13 is produced. For the purposes of our dataset, we have used a modified version of the FHM algorithm. We use the Fibonacci series to determine the squares and the dictionary entries. The pseudocode for this version is as given below. First the dictionary is built using selected points on squares S1,S2,S3,S5,S8 and S13, giving us 256 points. S1,S2,S3,S5,S8 and S13 are squares centered on the current anchor point with different sizes. See the pseudocode below for building the squares. Each point in the given dataset is then encoded as one of the dictionary points based on the nearest dictionary value that can represent this point.

Pseudocode for FHM Compression:

1) Pick a square size s, for the grid based on dataset
2) **for** square size s build static dictionary**do**
3) **for** square S1 **do** Pick 8 points with x or y distance from grid center as s **end**;
4) **for** square S2 **do** Pick 16 points with x or y distance from grid center as 2*s **end**;
5) **for** square S3 **do** Pick 24 points with x or y distance from grid center as 3*s **end**;
6) **for** square S5 **do** Pick 40 points with x or y distance from grid center as 5*s **end**;
7) **for** square S8 **do** Pick 64 points with x or y distance from grid center as 8*s **end**;
8) **for** square S13 **do** Pick 104 points with x or y distance from grid center as 13*s **end**;
9) **end**;
10) **for** each road segment **do**
11) Set the base point of the road segment as the current anchor point, i=0
12) i=i+1, Pick Pi, the next point on the road segment
13) **If** Pi is the first point outside any square Sn (n=1,2,...,8)
14) Get the closest point Mi in the dictionary which is closest to Pi
15) Encode Pi to Mi
16) Goto (12)
17) **Else**
18) Goto (12)
19) **end**;
20)**end**;

For example, consider the road segment in Figure 3. We provide the sequence of vectors created using ($X_i - X_{i-1}, Y_i - Y_{i-1}$) as inputs to the FHM-based compression. Figure 5 shows the dictionary and result of this compression. The square size chosen for this example is 1. The grid points for the first three squares that were generated based on the Fibonacci series are shown here. The differential vector $\vec{a}$ gets mapped onto the dictionary point 1, vector $\vec{b}$ gets mapped onto point 7, and vector $\vec{c}$ onto point 27 in the dictionary respectively.

3.3 Clustering-Based Compression (CBC) Methods

The clustering-based compression (CBC) method uses a clustering algorithm to generate a dictionary of the given

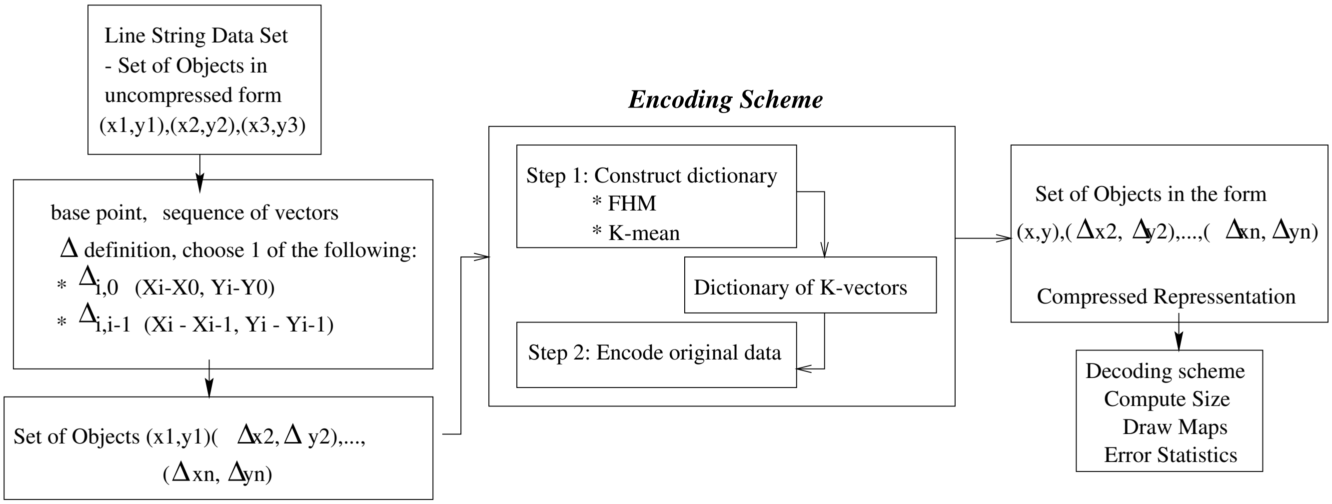

Figure 2: Dictionary Based Compression Technique

size. We have used the K-mean clustering [7] algorithm for dictionary design. K-mean clustering takes as input a fixed number (K) and generates that many clusters for the given dataset as output. The K-mean cluster centroids obtained becomes our dictionary entries. Based on this dictionary, we encode the vector dataset that we obtained earlier. Since each vector would now be assigned to a particular cluster, that vector would now be represented in terms of a reference to that cluster's centroid entry in the dictionary.

The encoding of each differential vector is the index of the spatially closest dictionary entry. For future decoding purposes, the dictionary is sent along with the encoded data. The pseudocode for this method is as shown below.

CBC Algorithm:
1) **for** Each road segment **do**
2) Separate base point and sequence of delta values
3) **end;**
4) **Do**
5) K-mean clustering on sequence of delta values (using clementine [2]), each cluster centroid is a dictionary entry
6) **for** Each original road segment **do**
7) Encode road segments using dictionary
8) **end;**
9) **end;**

The K-mean clustering algorithm provides a solution to the dictionary design problem stated in section 2. The objective function of K-mean clustering is the error of fit between differential vectors and centroids selected. It can be shown that it is the same as the objective of designing a dictionary for differential vector map compression with minimal spatial error. In Figure 4, a line segment is represented by its starting and ending points. The delta values are calculated as shown. After we cluster all the delta values $(\Delta {x_j}^e, \Delta {y_j}^e)$, the error of K-mean is calculated by

$$\sum_j (\Delta {x_j}^e - {x_{c_j}}^e)^2 + (\Delta {y_j}^e - {y_{c_j}}^e)^2$$

where (x_{c_j}, y_{c_j}) is the centroid of the cluster which contains $({x_j}^e - {x_j}^s, {y_j}^e - {y_j}^s)$, and the error in real space is calculated by

$$\sum_j ({x_j}^e - ({x_j}^s + x^e_{c_j}))^2 + ({y_j}^e - ({y_j}^s + y^e_{c_j}))^2$$

assuming that the difference vectors are relative to the first point in the line string. Since

$$\Delta {x_j}^e = {x_j}^e - {x_j}^s, \Delta {y_j}^e = {y_j}^e - {y_j}^s$$

a comparison of the first two equations reveals that the error resulting from the K-mean clustering approach equals the error in the real dataset.

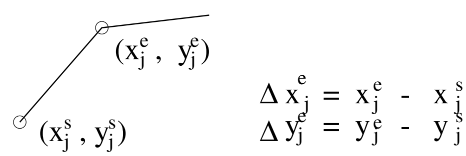

Figure 4: Line segment representation

K-mean clustering algorithms are often greedy and may not provide optimal solutions, i.e. optimal dictionary minimizing spatial errors of approximation. However, K-mean clustering can provide better dictionaries than those used in FHM.

LEMMA 1. *Given a fixed dictionary size, when the differential vectors are defined relatively to the first point of the line string, CBC always achieves equal or lower error than the modified FHM algorithm.*

Proof:
K-mean clustering starts from k random points as the k centroids. It clusters the dataset by assigning each point to the nearest centroid, recalculates the means of each cluster, and takes the means as the new centroids. If the total square error is decreasing, it iterates until the total square error is non-decreasing. When we compress and de-compress the data by CBC using the starting point of a road as the base point, each point $({x_j}^e, {y_j}^e)$ is represented as $({x_j}^s + x_{c_j}, {y_j}^s + y_{c_j})$ where $({x_j}^s, {y_j}^s)$ is the base point and (x_{c_j}, y_{c_j}) is the centroid of the cluster which contains $({x_j}^e - {x_j}^s, {y_j}^e - {y_j}^s)$. As we just showed, the error resulting from K-mean clustering is the same as the error produced by the compression

using the starting point of a road as the base point. By purposely choosing the initial K-centroids according to the entries in the FHM dictionary for a fixed grid size, and iterating through the K-mean clustering algorithm to achieve smaller total square error, we can guarantee that the CBC, which uses the starting point of a road as the base point, achieves lower error.

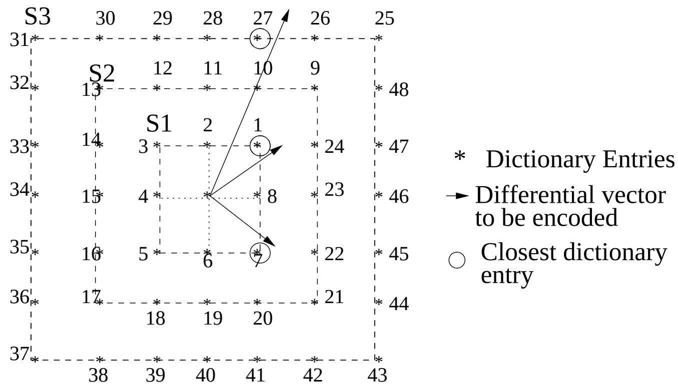

Figure 5: Example of Encoding by the FHM Algorithm

4. EXPERIMENT DESIGN AND RESULTS

The dataset, in shape file format, includes 47,014 road segments from the National Atlas of the United States of America, shown in Figure 6. The atlas contains the major roads and highways from the United States [4]. The shape files were converted to Arc/Info [1] coverage using the Arc command "SHAPEARC." Spatial attributes (coordinates) of the coverage were then extracted in a text file using the Arc command "UNGENERATE". The dictionary-based map compression methods were applied on this extracted coordinates dataset by first obtaining the delta values for each of the road segments. For this particular dataset, we obtained 409,964 delta values, the distribution of which are shown in Figure 7. K-mean clustering was then done on these delta values using Clementine [2] with the value of K set to 256 so as to build a dictionary of 256-mean entries. The original dataset was then encoded using the dictionary built.

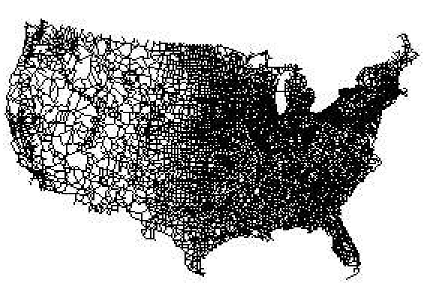

Figure 6: US Road Map
[4]

In the experiment design, shown in Figure 1, a given vector map is first decomposed to geometric data and topological data. The geometry data is then converted to delta

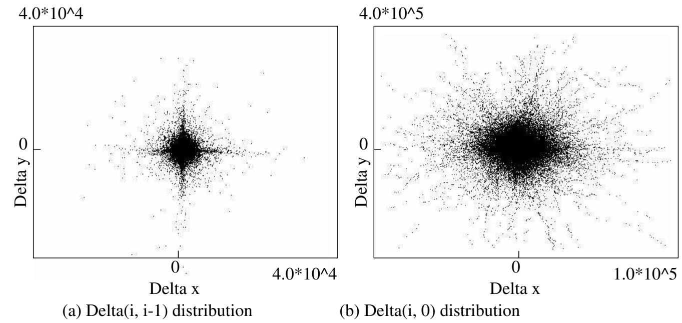

Figure 7: Benchmark Vector Map Vector Distribution

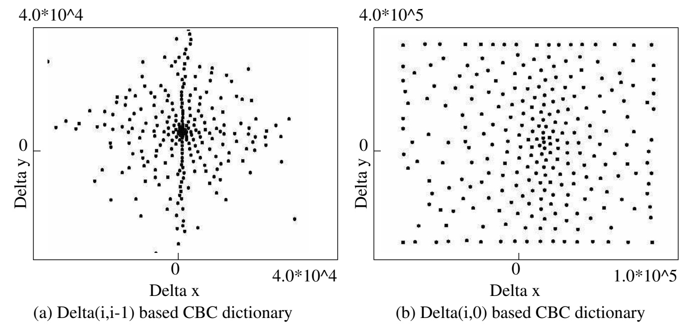

Figure 8: CBC Dictionaries

vectors. There are two delta definitions: one related to the first node of the road segment and one related to the previous node of the road segment. FHM or CBC algorithms could be used to design the dictionary using the delta values. Then the dictionary is used to compress the original dataset. The final step is to evaluate the compression scheme by computing the error and visualizing the de-compressed dataset compared with the original dataset.

4.1 CBC Based Compression Result

The major difference between our technique and the FHM algorithm is that we use the K-mean clustering technique to produce the dictionary entries, while the FHM uses a predefined dictionary. Figures 7 and 8 show the original delta vector distributions and CBC dictionaries. We note that the dictionaries designed by CBC closely mimic the actual distribution of the differential vectors.

Defining differential vectors using A-$\Delta_{i,i-1}$, we encoded the dataset using the CBC algorithm, and then decoded the datasets. In Figure 9, we superimpose the decoded dataset on the original dataset and display a typical region of the map using ESRI's ArcExplorer 3.1. We observe that for this particular dataset, the decoded map matches closely with the original map.

4.2 The Impact of Delta Choice on FHM

Figure 10 gives the root mean square error of FHM for both delta definitions with the dictionary grid size increasing. In both delta definitions, there is an optimal grid size choice. The optimal grid size choice for the delta vector defined based on the first point($\Delta_{i,0}$) is much larger than the that for the delta vector defined based on the previous point($\Delta_{i,i-1}$). This may be because the lengths of the

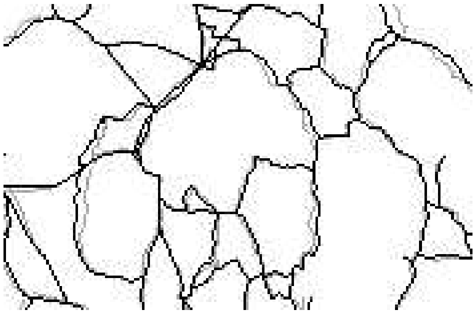

Figure 9: Map Produced by CBC(light color) and the Original(dark color)

former delta vectors are generally larger than the lengths of the later delta vectors. The figure also shows that FHM achieves much lower error with the $\Delta_{i,0}$ definition, than with the $\Delta_{i,i-1}$ definition. This may be due to the large variation of vector length in $\Delta_{i,0}$. We note that the optimal grid size could be related to the average length of the delta vectors.

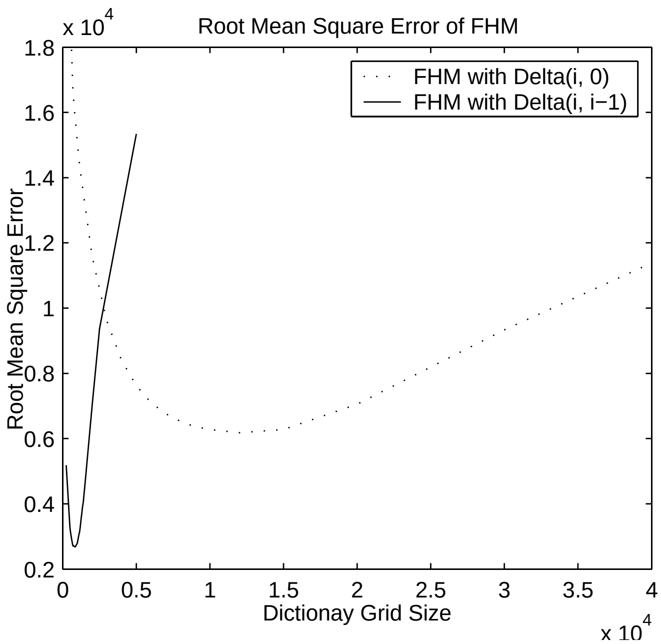

Figure 10: Errors of FHM

4.3 The Impact of Delta Choice on CBC

Table 1 shows the errors of CBC with different delta definitions. The K-mean error is different from the root square error of CBC with the delta definition based on the previous point because the errors accumulate when going away from the starting point. The CBC achieves lower error when the delta definition is based on the previous point in spite of the error accumulation in decoding, possibly because the number of line segments in each linestring is small(average of 8). With the number of shape points per road segment increasing, error accumulation might increase for CBC with the delta definition based on the previous point.

Delta Definition	Root Mean Square Error
$\Delta_{i,i-1}$	2818
$\Delta_{i,0}$	3267

Table 1: CBC Root Mean Square Error Values

4.4 Comparison of FHM and CBC

The root mean square error of FHM and CBC are compared in Figure 11 and Figure 12 for $\Delta_{i,0}$ and $\Delta_{i,i-1}$. CBC achieves lower error than FHM for $\Delta_{i,0}$ as predicted by Lemma 1. For the $\Delta_{i,i-1}$, FHM achieves lower error at some grid sizes. If we provide the dictionary entries generated by FHM as initial centroids to the K-mean algorithm in CBC, then CBC may find a dictionary with lower error. In addition, CBC is simpler to use since it does not require guessing and tuning parameters such as grid size.

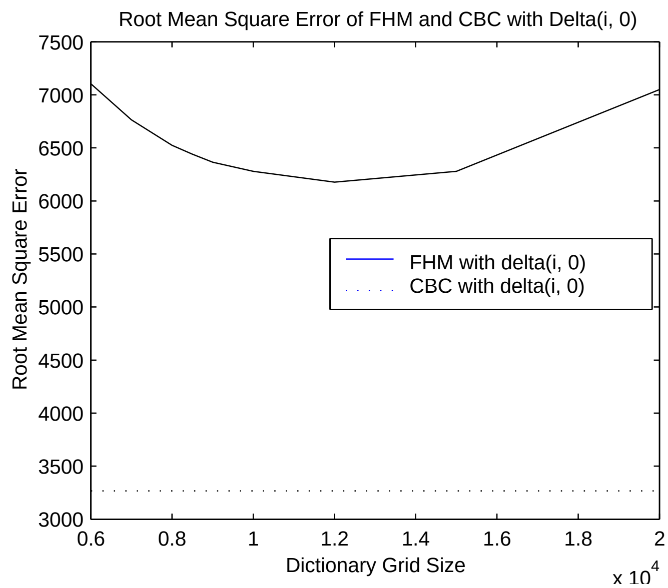

Figure 11: Errors of FHM and CBC with Delta(i, 0)

5. CONCLUSION AND FUTURE WORK

In this paper we have explored the problem of designing dictionaries for dictionary-based vector map compression with the objective to minimize the error. We have proposed the use of clustering algorithms to produce an adaptive dictionary based on a given dataset. We have also formally shown through a lemma that this proposed dictionary construction approach yields lower errors of approximation than the conventional fixed dictionary techniques used by the FHM algorithm, when using vectors defined with respect to the first point of each line-string.

In future work we would like to further the application of clustering algorithms to vector map compression. We would like to investigate the impact of different clustering algorithms on compression accuracy. The general purpose clustering algorithm aims at minimizing the total square error. The clustering algorithm needs to be modified to handle cases where an error threshold or map accuracy [3] is spec-

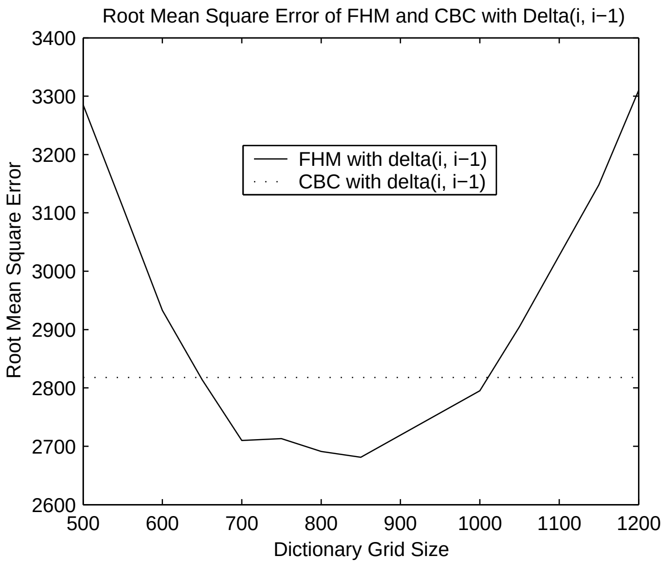

Figure 12: Errors of FHM and CBC with Delta(i, i-1)

ified as a constraint. We would also like to investigate the trend of compression ratio by fixing error bounding.

Acknowledgments

We thank Raju Vatsavai, Xiaobin Ma, Hui Xiong, and Pusheng Zhang of the University of Minnesota for their valuable feedback on earlier versions of this paper.

6. REFERENCES

[1] Arc/Info 8. *http://www.esri.com/software/arcview/index.html.*

[2] Clementine. *http://www.spss.com/clementine/.*

[3] Map Accuracy Standard. *http://mac.usgs.gov/mac/isb/pubs/factsheets/fs17199.html.*

[4] National Atlas of the United States of America . *http://www.nationalatlas.gov/.*

[5] S. Chawla, S. Shekhar, and U. Ozesmi. Extending Data Mining for Spatial Applications: A Case Study in Predicting Nest Locations. In *Proc. Int. Confi. on 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2000)*, 2000.

[6] D. Douglas and T. Peucker. An algorithm for the reduction of the number of points required to represent a digitized line or its character. *The Canadian Cartographer*, 10, 1973.

[7] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[8] C. C. Lu and J. G. Dunham. Hightly Efficient Coding Schemes for Contour Lines Based on Chain Code Representations. *IEEE Transactions on Communications*, 39(10):1511–1514, Mar. 1991.

[9] D. Salomon. *Data Compression: the Complete Reference.* Springer-Verlag, 2nd edition, 2000.

[10] S. Shekhar and S. Chawla. Spatial Databases; A Tour. *Prentice Hall, ISBN 0-13-017480-7*, 2003.

[11] S. Shekhar and Y. Huang. Co-location Rules Mining: A Summary of Results. In *Proc. Spatio-temporal Symposium on Databases*, 2001.

[12] S. Shekhar, Y. Huang, and J. Djugash. Dictionary Design Algorithms for Vector Map Compression(Abstraact). In *Procedings of Data Compression Conference*, 2002.

[13] S. Shekhar, C. Lu, and P. Zhang. Detecting Graph-based Spatial Outliers: Algorithms and Applications. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.

[14] S. Shekhar, T. Yang, and P. Hancock. An Intelligent Vehicle Highway Information Management System. *Intl Jr. on Microcomputers in Civil Engineering (Publisher: Blackwell Publishers), 8(3)*, 1993.

[15] K. Thappa. Automatic line generalization using zero-crossings. *Photogrammetric Engineering and Remote Sensing*, 54(4), Apr. 1988.

[16] P. W. Wong and J. Koplowitz. Chain Codes and Their Linear Reconstruction Filters. *IEEE Transactions on Information Theory*, 38(2):268–280, Mar. 1992.

[17] M. Worboys. *GIS: A Computing Perspective.* Taylor and Francis, 1995.