COMPRESSION OF SPATIAL DATA

Roman M Krzanowski
NYNEX Science & Technology
500 Westchester Ave.
White Plains , NY

ABSTRACT

The objective of this study was to quantify the compressibility of selected spatial data sets: USGS DEM 3 arc-sec, ETAK, and TIGER/Line. The study serves several purposes: it provides the detailed description of the structure of spatial data sets from the perspective of the compression process (using n-gram statistics and entropy); compares the effectiveness of different compression methods (using compression rates), and provides the recommendations on the use of compression methods for the compression of spatial data for both UNIX and DOS operating systems. Three main conclusions are reached in this paper: the compression rates for spatial data sets may be predicted from their entropy; the compression rates for a given type of spatial data remain stable for different instances of those data (exception are DEM data); and currently available compression programs can achieve between 80 percent and 90 percent compression rates on spatial data.

INTRODUCTION

This paper presents a comprehensive study of the compression properties of spatial data. The size of spatial data sets quite frequently exceeds 10 megabytes (Mb). Transfer of such data in large volumes requires either high capacity storage media or high capacity data networks. In either case, the transfer of data is greatly enhanced if the transferred data are efficiently "packed" (that is compressed) (Storer, 1992). Efficient packing of data requires knowledge of data packing properties which are very data dependent * . A review of the literature has revealed that most of the compression studies reported have been concerned with the compression of binary images, sound, voice, or textual files (Held, 1991; Nelson, 1991b; Nelson, 1992; Storer, 1988; Welch, 1984). To the author's knowledge no systematic study of the compression of spatial data has been published. This paper is an attempt to provide such information .

The results of this study may be used in planning data networks, designing of distributed data bases; planning storage space requirements for spatial data bases; and defining of the requirements for secondary storage media. The intended audience includes vendors of spatial data, and any federal, state or local, agencies dealing with spatial data transfer, storage, and distribution.

The next section of this paper introduces fundamental concepts and definitions related to data compression and also reviews modern compression algorithms. Then, findings of a series of experiments establishing the compression characteristics of selected spatial

* "It is clear that the performance of each of ... (compression) ... methods is dependent on the characteristics of the source..." (Lelewer & Hirschberg, 1988, p.288).

data are presented. Final sections review the results of those experiments, and also provide guidelines for the selection of compression methods for spatial data.

FUNDAMENTAL CONCEPTS

Basic definitions

Data compression is the process of encoding a set of data D into a smaller set of data ΔD. It must be possible to decode the set ΔD into the set D -- or to its approximation. Compression methods can be "lossless" and "lossy". "Lossless" methods compress a set of data and thereafter, decompress it into exactly the same set of original data. "Lossy" data compression converts the set of data into a smaller set from which an approximation of the original set may be decoded. "Lossy" data compression is used for the packing of images, speech, or sound, and is appropriate for the compression of data when their accuracy can be compromised. As the accuracy of the spatial data cannot be sacrificed in the compression process, "lossy" compression methods are not suitable for the compression of spatial data as defined in this study. The reminder of this paper will concentrate on "lossless" compression methods.

In this paper a data set processed by the computer is synonymous with a source message. A source message is composed of words from a source alphabet* . Computer processing of a data set involves the process of coding of a source message. Coding is a mapping of a source message (words from a source alphabet) into the code words (words from an code alphabet). A simple example of coding is the replacement of letters from an English alphabet by the 7-bit ASCII code from $\{0,1\}$ alphabet.

The measure of the efficiency of coding (or the message information content) of a message (word) a_i in bits is $-\log_2 p(a_i)$ where $p(a_i)$ is a probability of occurrence of the message a_i in the source message. The average information content of a source message is called entropy (H) and is expressed as:

$$H = \sum_{i=1}^{n} [-p(a_i)\log_2 p(a_i)] \tag{1}$$

In terms of the coding efficiency, the entropy gives the lower bound on the number of bits necessary to encode the source message (Lelewer & Hirschberg, 1987). Number of bits in the coded message above its entropy is called the redundancy. The amount of redundancy *(R)* in the message is expressed as:

$$R = \sum p(a_i)l_i - \sum [p(a_i)\log p(a_i)] \tag{2}$$

where l_i is the length of the code word representing the message a_i .

In most cases, uncompressed coding of data creates redundancy** . The most obvious form of redundancy are repeated patterns of words in the source message. Those patterns are called grams. The existence of repeated patterns (a_i) in the message is determined by

* Source message A is an ensemble of messages (words) $a_i \in A, A = \{a_i,\ldots,a_n\}$.

** Welch (1984) distinguishes four types of redundancy that affect compression: character distribution, character repetition, high-usage patterns, and positional redundancy.

the compilation of "dictionaries" of the patterns with their frequencies (probabilities of occurrence - $p(a_j)$) *. Patterns or grams may be of 1,2 ,3 or higher order representing one, two, three or more character patterns. 1-order (also called 0-order) or 1-grams ignore the dependency of a pattern on preceding or following words. For any meaningful source message this is an unrealistic assumption. Yet, despite their simplicity, frequencies (and entropy) of 1-order patterns provide valuable information about the source message (to be demonstrated later). The grams of order 2 (2-grams),3 (3-grams) or higher provide frequencies (and entropies) of the longer patterns. Longer patterns quantify the dependency between preceding and following words. Figure 1 demonstrates some of redundancies encountered in spatial data files.

The overall efficiency of any compression method is measured by its compression rate. There are several measures of the compression rate (Lelewer & Hirschberg, 1988; Nelson, 1991b) **. In this paper the compression rate is expressed as a ratio of the size (in bytes) of compressed to uncompressed source message:

$$c = \frac{fs}{fsc} * 100\% \tag{3}$$

where ***fs*** is a size of an uncompressed source message in bytes, ***fsc*** is a size of a compressed source message in bytes, and ***c*** is the compression ratio (Nelson , 1991b).

Compression algorithms and their implementations

Compression is a two step process consisting of modeling and coding (Hirschberg & Lelewer, 1992; Nelson, 1992). Modeling step creates the representation of the source message. The coding step generates the compressed message based on the model. Models may be either statistical or dictionary based.

In "statistical" modeling, the frequency of occurrence of words in the source message is first calculated. Then, using this frequency the new codes that use fewer bits than the original codes are assigned to words. Compression is achieved by the difference between the size of the original code words and the new code words. An early example of the statistical coding is Morse's code optimized for the English language. The important parameter of the compression based on statistical models is the order of the grams for which the frequency of occurrence is calculated. A detailed explanation of statistical modeling methods is offered in Lelewer & Hirschberg (1988). The implementation details are also explained in Nelson (1992).

In dictionary-based methods the coding does not produce the smaller code words but it produces pointers to the patterns encountered in the source message. The source message is scanned using the "window" of predefined size. A dictionary of patterns in the window is created as the source message is scanned and pointers to those patterns are inserted in the coded message. Each pointer contains the index of the pattern in the dictionary and the first character not in the pattern. The dictionary of patterns is updated as the new patterns are scanned till the maximum size of the dictionary or patterns is reached. Compression is achieved when frequent long patterns are substituted with shorter pointers

* Source message with purely random patterns cannot be compresssed (Strorer 1988).

** Compression rate is a compression yielded by a coding scheme. In addition to the measure adopted in this study compression rate may be measured by the ratio of the average message length to the average code word length (Lelewer & Hirschberg, 1988).

to dictionary entries. The important parameters of the dictionary based methods are: size (in bytes) of the scanning window, the size (in bites) of the index, and the size of the largest pattern held in the dictionary. The detailed explanation of algorithms of dictionary compression methods is offered in Lelewer & Hirschberg (1988). The implementation details are explained in Nelson (1992).

Both statistical and dictionary-based compression methods can be either static or adaptive (dynamic). Static methods do not adjust their parameters to the type of the source message; adaptive methods change their parameters in the response to the properties of the processed message.

The statistical compression methods use Huffman, Shannon-Fano, or arithmetic coding (Held, 1991; Lelewer & Hirschberg, 1988; Howard & Vitter, 1992; Nelson, 1991a) and are uncommonly the primary coding algorithms found in the commercial compression software. Rather, dictionary-based compression methods, which use LZ77 or LZ78 algorithms or their derivatives (LZSS or LZW) are employed in the creation of most of the modern compression software (Nelson 1992) * .

Compression methods based on statistical models are limited by the size of the model of the source message that increases with the order of the model. Huffman based coding methods loose efficacy as they use only whole bits for code words (frequently, information may be represented by a fraction of a bit - in a sense of the information content) (Hirschberg & Lelewer, 1992). Arithmetic methods, while very efficient, require large computer resources and are generally very slow (Nelson, 1992). The most efficient compression methods currently used are dictionary-based, an observation which is confirmed by the prevalence of those methods in most commercial implementation (some drawbacks of those methods are explained in Lelewer & Hirschberg (1988)).

Compression rates achieved by compression programs may be as high as 98 percent (Lelewer & Hirschberg, 1988) for the source specific compression programs. On average, the reported compression rates for English texts range from 50-60 percent (Lelewer & Hirschberg, 1987); 40 percent for Huffman based compression (Nelson, 1992); and 40 - 60 percent for dictionary based methods. As noted earlier, no systematic analysis of compression rates was reported for spatial data.

COMPRESSION TESTS

Methodology

Compression algorithms. This study evaluated the commercial compression packages listed in Table 1 as well as generic compression methods listed in Table 2. Compression packages COMPRESS, GZIP, ARJ, PKZIP are available either as the part of OS, from Internet sites (GZIP, 1993; PKZIP, 1993), or through software vendors. Compression software listed in Table 2 is available in Nelson (1992). Compression methods based on statistical modeling (Huffman; Arithmetic coding) have been tested for comparison purposes only. No current production compression software uses either as their primary compression method.

* A different taxonomy of compression algorithms has been proposed by Lelewer & Hirschberg (1988).

COMPRESS utility implements LZW algorithm based on LZ78 which is an extension of LZ77 algorithm (Nelson, 1992; Welch, 1984). GZIP is a variation of LZ77 with the elimination of duplicate strings and use of Huffman coding for compression of indexes (GZIP 1992). ARJ, PKZIP both implement LZ77 based algorithms. Huffman compression, adaptive Huffman compression, and arithmetic compression programs tested in this study are based on 1-order models. LZSS program tested in this study is an implementation of LZSS extension to LZ77 with 4096 -byte window size, 12 bit index to the window, and up to 17 bytes of the coded pattern length (Nelson, 1992). LZW program tested here uses 12 bit fixed code length and is an extension of LZ78 (Nelson, 1992; Welch, 1984).

Spatial data. Spatial data sets tested in this study are listed in Table 3. The selected spatial data usually constitute the fabric of the land information systems. Some of these spatial data (USGS DEM *) are already available on Internet, others (ETAK ** , TIGER***) are distributed on CD-ROMs.

Statistical Measures. In this study the compressibility of the data sets was assessed using the entropy defined by formula (1) , redundancy defined by the formula (2), and the frequency of grams for f 1-, 2-, and 3-, orders . The compression rates were evaluated using the formula (3). The compression rates for Tiger/line files (T1,T2,T3) were averaged for all of the files in the set (records 1 to 8, and a to r).

Results of experiments

The results of the analysis of the structure of spatial data - entropy of 1-,2-,and 3- order, the statistics of three most frequent 1-order grams, and the redundancy - are given in Tables 4, 5, and 6 respectively. Table 7 reports the compression rates achieved with the selected compression methods and calculated using the formula (3) .

CONCLUSIONS

The following observations can be made about the compression properties of the tested spatial data sets:

- Compression rates for spatial data are above those of English text, or program files;
- entropies and redundancies of the same order, for a given type of spatial data, are similar (see Table 4 and Table 6);
- average entropies of order 1, 2, and 3 (Table 8) for ETAK and TIGER/Line data sets are either above (1-gram) or below (2-gram, 3-gram) entropy of DEM data sets. This suggests the similar coding structures and coding efficiencies of ETAK and TIGER /Line;

* Format of USGS DEM 250 data sets is defined in Digital Elevation Models, (1992), Data Users Guide 5. US Department of the Interior, U.S. Geological Survey, Reston, Virginia.

** Format of ETAK data is described in ETAK (1993), MapBase File Definition, File Format version 2.0-2.2, ETAK, The Digital Map Company.

*** Format of TIGER /Line data is defined in TIGER/Line Precensus Files,(1990), Technical Documentation, U.S. Department of Commerce, Washington, D.C.

- the most frequent 1-gram (Table 5) in tested data sets is a white space character (ASCII 32), it constitutes from 44 to 57 percent of characters in the data sets (D3 excluded);

- the statistics of 1-grams are very similar for all of the tested data sets. Most of the grams have a frequency below 4 percent (i.e. the most frequent gram has a frequency above 40 percent, the next one has a frequency above 4 percent, and the rest has a frequency below 4 percent);

- a data format that maps the actual locations in space to its file format (DEM) has a lot of redundancy. This is reflected in the high compression rates for DEM data set.

The following observations can be made about the compression methods evaluated in this project (Table 7):

- Compression methods based on statistical modeling (Huffman, Adaptive Huffman, Arithmetic coding) are inferior to dictionary-based methods (LZ77 and LZ78 and their derivatives). Dictionary-based compression methods demonstrated from 10 percent to 20 percent greater compression rates;

- regardless of the packing method studied, the compression rates for a given spatial data type do not vary significantly (the sole exception being DEM data);

- commercial, dictionary-based compression methods yield the compression rates above the redundancy calculated from 3- grams.

RECOMMENDATIONS AND SUGGESTIONS

The following presents recommendations for the compression of spatial data, utilization of compression programs for the packing of spatial data and concludes with suggestions for further research in this area:

- Knowledge of the n-order entropies and related redundancies may be used for the prediction of the compression rate for a given data type: compression rates with compression methods based on the statistical model are usually close to the 1-order redundancy, compression rates with compression methods based on dictionaries exceed the 3-order redundancy by 10 to 15 percent (Table 9 and Table 10);

- when using commercial dictionary-based compression methods one may expect compression rates of spatial data sets to be from 84 percent to 90 percent. In rare cases (for specific types of formats) it may be as high as 99 percent;

- the best compression rate in this study was demonstrated by GZIP compression software for UNIX, and ARJ and PKZIP compression software for DOS;

- future studies should be carried out into the time aspect of the data packing process as the amount of time taken by the compression program varies significantly from an implementation to another (time was not recorded in this study except when it exceeded an arbitrary limit of 30 minutes - see Table 7);

- future studies into the compressibility of spatial data sets should concentrate on the analysis of longer than 3- grams, and on the specific features of spatial data (use of floating point numbers, absolute coordinate system);

- observations and conclusions reported in this study necessarily reflect the amount and type of data tested, studies should be carried out on the larger amount of spatial data sets before findings of this study could be safely generalized.

REFERENCES

GZIP 1993 , Algorithm documentation, ftp.cso.uiuc.edu.

Held, G. 1991 , Data Compression, New York: John Wiley & Sons, Ltd.

Hirschberg, D.S. & Lelewer, D.A. 1992 , Context Modeling for Text Compression, in Image and Text Compression, J.A. Storer (ed), Kluwer Academic Publishers: London, pp.113-144.

Howard, P.G. & Vitter, J.S. 1992 , Practical Implementations of Arithmetic Coding, in Image and Text Compression, J.A. Storer (ed), Boston: Kluwer Academic Publishers, pp. 85-109.

Lelewer, D.A. & Hirschberg, D.S. 1988 , Data Compression, ACM Computing Surveys, 19(3), pp. 261-296.

Nelson, M.R. 1991a , Arithmetic Coding and Statistical Modeling, Dr.Dobb's Journal, 2, pp. 16-29.

Nelson, M. R. 1991b, Data Compression Contest Results, Dr. Dobb's Journal, 11, pp. 62-64.

PKZIP 1993, Algorithm documentation, ftp.cso.uiuc.edu.

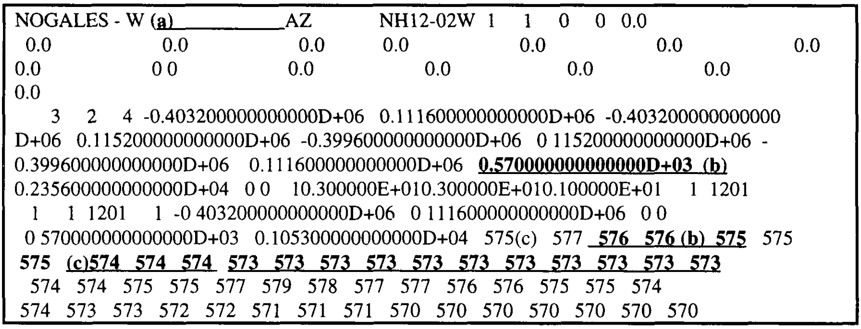
Nelson, M. 1992, The Data Compression Book, New York: M&T Publishing, Inc.

Storer, J.A. 1988, Data Compression, Methods and Theory, Maryland: Computer Science Press, Inc.

Storer, J.A. 1992, Introduction, in Image and Text Compression, J.A. Storer (ed), Boston: Kluwer Academic Publishers, pp.v-viii.

Thomas, K. 1991, Entropy, Dr. Dobb's Journal, 2, pp. 32-34.

Welch, T.A. 1984 , A Technique for High Performance Data Compression, Computer, June, pp. 8-18.



```
NOGALES - W (a)             AZ          NH12-02W  1    1    0    0   0.0
 0.0             0.0             0.0        0.0             0.0             0.0             0.0
0.0             0 0             0.0             0.0             0.0             0.0
0.0
    3    2    4  -0.403200000000000D+06   0.111600000000000D+06  -0.403200000000000
D+06   0.115200000000000D+06  -0.399600000000000D+06   0 115200000000000D+06  -
0.399600000000000D+06    0.111600000000000D+06   0.570000000000000D+03  (b)
0.235600000000000D+04   0 0    10.300000E+010.300000E+010.100000E+01     1  1201
  1    1  1201     1  -0 403200000000000D+06   0 111600000000000D+06   0 0
 0 570000000000000D+03   0.105300000000000D+04   575(c)   577  576   576 (b)  575   575
 575   (c)574   574   574   573   573   573   573   573   573   573   573   573   573   573
  574   574   575   575   577   579   578   577   577   576   576   575   575   574
 574   573   573   572   572   571   571   571   570   570   570   570   570   570   570
```

Figure 1. Example of redundancies in Spatıal Data file (DEM) - (a) white spaces; (b) use of numbers (limited alphabet); (c) repeated patterns.

Compression Software	**Operating System**	**Compression method**
COMPRESS*	UNIX	Dictionary-based (LZW)
GZIP*	UNIX	Dıctionary-based (LZW)
ARC,PKARC	DOS	Dictionary-based (LZ78)
ARJ*	DOS	Dictıonary-based/ (LZ78) Huffman
LHarc	DOS	Dictionary-based
PKZIP*	DOS	Dictionary-based (LZ78)

(*)- software tested in this study.

Table 1. Most common compression programs and related compression algorithms.

COMPRESSION ALGORITHM	**COMPRESSION METHOD**
Huffman	statistical
Adaptive Huffman	statistical
Arithmetic Codıng 0	statistical
LZSS	dıctionary-based
LZW	dictionary-based

Table 2 . Compression algorithms tested in this study.

CODE	DATA TYPE	DATA FILE	SIZE [byte]
D1	USGS DEM	Nebraska-w	9,840,640
D2	250 3arc-sec	Nogales-w	9,840,640
D3		Noyo-canyon-e	9,840,640
E1	ETAK	dnv_co.mbs	80,253,440
E2		ftc_co.mbs	41,170,688
E3		mhn_cy.mbs	19,004,772
	TIGER/line census	state 36	
T1	county 003	record 1...r	9,197,728
T2	county 071	record 2. .r	22,395,596
T3	county 111	record 3...r	15,296,530

Table 3. Data sets used in the study and their respective sizes ın bytes.

DATA SET	1-gram	2-gram	3-gram
E1	2.66	4.31	5.77
E2	2.64	4.31	5 75
E3	2.78	4.49	5.95
D1	2.28	4.05	5 42
D2	2.59	4.89	6.85
D3	0.63	1.22	1.75
T1	2.78(1.94-3.4)*	4.20(1.94-3.40)	5.61(4.15-7 3)
T2	2.77(2.0-3.5)	4.38(2.0-3.51)	5.52(5.0-7.51)
T3	2.74(2.0-3.5)	4.35(4.0-5.7)	5.51(4.9-7.5)

(*)- for TIGER line files the reported entropy of grams is the average entropy of grams for all of the files ın the given data set and the minimum and maximum in the set.

Table 4. Entropy of 1- 2- and 3-grams for tested data sets.

DATA SET	1-grams
E1	44%(32) 24%(48) 4%(49)*
E2	42%(32) 26%(48) 4%(49)
E3	40%(32) 25%(48) 5%(49)
D1	54%(32) 13%(52) 4%(53)
D2	47%(32) 12%(49) 4%(50)
D3	84%(32) 15%(48) 7%(41)
T1	58%(32) 6%(48) 4%(51/50)**
T2	55%(32) 6%(48/45) 4%(51)
T3	57%(32) 7%(49) 4%(48)

(*) - percentage of 1-grams(ASCII code of 1-gram); (**) - statistics for record type 1.

Table 5. Statistics of the most frequent 1-grams

DATA SET	1-order	2-order	3-order
E1	66%	73%	76%
E2	67%	73%	76%
E3	65%	72%	75%
D1	71%	74%	77%
D2	67%	69%	71%
D3	92%	92%	92%
T1	65%	73%	76%
T2	65%	72%	77%
T3	65%	73%	77%

Table 6. Redundancy for tested data sets for 1-, 2-, 3-order entropy.

DATA SET	COM PRES S	GZIP	HUF	AHUF	AR-0	LZSS	LZW	ARJ	PKZI P
E1	84%	88%	f*	f	f	78%	72%	88%	88%
E2	84%	88%	f	f	f	78%	83%	88%	88%
E3	84%	89%	f	f	f	80%	74%	88%	88%
D1	89%	91%	70%	71%	73%	81%	78%	91%	91%
D2	81%	82%	66%	67%	70%	71%	67%	85%	87%
D3	99%	99%	86%	86%	92%	88%	99%	99%	99%
T1(**)	85%	90%	65%	66%	65%	79%	80%	90%	90%
T2	84%	90%	64%	64%	64%	72%	77%	90%	90%
T3	85%	90%	64%	64%	64%	79%	79%	90%	90%

(*) f - failed to compress in 30 min; (**) - the average for all files in the set; HUF- Huffman compression order 0; AHUF- adaptive Huffman compression order 0; AR-0 - arithmetic compression order 0; LZSS - LZ77 based; LZW - modified LZ78;

Table 7. Compression rates for tested algorithms and methods.

	E	D*	T
1-gram	2.69	2.43	2.76
2-gram	4.37	4.47	4.26
3-gram	5.82	6.13	5.53

(*) - D3 excluded; E - ETAK ; D - DEM; T - TIGER .

Table 8. Average entropy for tested data sets.