Computers & Geosciences 100 (2017) 41–45

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

# An improved lossless group compression algorithm for seismic data in SEG-Y and MiniSEED file formats

Huailiang Li[a,*], Xianguo Tuo[a,b,c], Tong Shen[b,c], Mark Julian Henderson[a], Jérémie Courtois[a], Minhao Yan[a]

[a] Fundamental Science on Nuclear Wastes and Environmental Safety Laboratory (Southwest University of Science and Technology), Mianyang 621010, China

[b] Sichuan University of Science & Engineering, Zigong 643000, China

[c] State Key Laboratory of Geohazard Prevention & Geoenvironmental Protection (Chengdu University of Technology), Chengdu 610059, China

ARTICLE INFO

*Keywords:*
Lossless compression
Group compression
Seismic data
SEG-Y
MiniSEED

ABSTRACT

An improved lossless group compression algorithm is proposed for decreasing the size of SEG-Y files to relieve the enormous burden associated with the transmission and storage of large amounts of seismic exploration data. Because each data point is represented by 4 bytes in SEG-Y files, the file is broken down into 4 subgroups, and the Gini coefficient is employed to analyze the distribution of the overall data and each of the 4 data subgroups within the range [0,255]. The results show that each subgroup comprises characteristic frequency distributions suited to distinct compression algorithms. Therefore, the data of each subgroup was compressed using its best suited algorithm. After comparing the compression ratios obtained for each data subgroup using different algorithms, the Lempel-Ziv-Markov chain algorithm (LZMA) was selected for the compression of the first two subgroups and the Deflate algorithm for the latter two subgroups. The compression ratios and decompression times obtained with the improved algorithm were compared with those obtained with commonly employed compression algorithms for SEG-Y files with different sizes. The experimental results show that the improved algorithm provides a compression ratio of 75–80%, which is more effective than compression algorithms presently applied to SEG-Y files. In addition, the proposed algorithm is applied to the miniSEED format used in natural earthquake monitoring, and the results compared with those obtained using the Steim2 compression algorithm, the results again show that the proposed algorithm provides better data compression.

## 1. Introduction

Three-dimensional (3D) engineering seismic acquisition and high-precision 3D seismic acquisition are being increasingly utilized in the gas and petroleum exploration industry owing to the constant expansion of oil and gas exploration and the increasing complexity of exploration targets, which is accompanied by an increasing scale and complexity in the layout of seismic sources and receivers, and the acquisition of an ever growing amount of data (Hanchuang, 2015). For example, in the 3D exploration task of Daqing Peace Ranch in 2013, the data generated by the single shot was about 140 MB, and the amount of data collected daily was about 100–140 GB, the amount of data collected daily in two-dimensional exploration missions is between 5 and 10 GB (Xiuwei, 2014). Therefore, it is imperative to employ data compression techniques for reducing the size of the acquired data files. The practical application of compression techniques to seismic acquisition data will improve processing efficiency and provide the following economic benefits. (1) After compression, seismic acquisition data requires less storage space, which reduces costs indirectly. (2) Under fixed communications hardware conditions, data compression is a practical and effective solution for timely data transmission over communication channels or networks, which can reduce transmission costs and improve transmission efficiency.

Data compression techniques fall under two broad categories, namely, lossless and lossy compression. Lossless compression is also known as lossless coding, entropy coding, and information holding coding, in which the compressed data can be restored to its original state by decompression (Mielikainen and Huang, 2012). The lossless category is represented by numerous compression algorithms such as Shannon coding (Shannon, 2001), Huffman coding (Huffman , 1952), run-length encoding (Robinson and Cherry, 1967), arithmetic coding (Witten et al., 1987), the Lempel-Ziv 1977 (LZ77) algorithm (Ziv and Lempel, 1977), the Lempel-Ziv 1978 (LZ78) algorithm (Ziv and Lempel, 1978), and the Lempel-Ziv-Welch (LZW) algorithm (Welch,

* Corresponding author.
*E-mail address:* li-huai-liang@163.com (H. Li).

http://dx.doi.org/10.1016/j.cageo.2016.11.017
Received 26 October 2016; Accepted 29 November 2016
Available online 01 December 2016

1984). Nowadays, most commonly employed lossless compression algorithms have evolved from a combination of the algorithms described above with some other particular algorithms. Examples of such algorithms are the Lempel-Ziv-Markov chain algorithm (LZMA) (Leavline and Singh, 2013), which developed from LZ77, the Deflate algorithm (Harnik, 2014), which is a combination of LZ77 and Huffman coding, and the BZip2 algorithm (Szecówka and Mandrysz, 2009), which is a hybrid algorithm that combines the Burrows-Wheeler transform (Crochemore et al., 2015), run-length encoding, and Huffman coding.

At present, a mainstream lossless compression algorithm denoted as Steim2 has been employed in natural earthquake monitoring (Mariotti and Utheim, 2006; Canabrava, 2004), and has been widely used in network transmission and storage for miniSEED data (Augliera et al., 2011; Ringler et al., 2015). However, no lossless compression algorithm has been developed specifically for the standard format files of engineering seismic data (e.g., SEG-Y, which is the most widely used seismic data format), although Rubin et al. developed a lossy compression algorithm for wireless seismic data acquisition and storage (Rubin et al., 2016; Lindstrom et al., 2016). Therefore, we present a high compression ratio lossless data compression algorithm suitable for the SEG-Y format.

## 2. Analysis of SEG-Y and miniSEED data formats

### 2.1. SEG-Y data format and distribution

(1) Data format. SEG-Y is the most commonly employed format in seismic exploration, which includes volume header data, trace header data, and trace data, as presented in Table 1.

Among these segments, the total length of the volume header data is 3600 bytes, and the first 3200 bytes are Extended Binary Coded Decimal Interchange Code (EBCDIC) character encoding, which is used to store some descriptive information about the seismic data volume. The second 400 bytes store some of the key SEG-Y file information such as the data format, measurement units, sampling points, and the sampling interval. Most of these fields are stored in fixed byte positions. For example, the recording traces of every single shot are recorded in byte positions 3213–3214, the sampling interval is stored in 3217–3218, and sampling points are recorded in 3221–3222. Trace header data generally includes a trace sequence number (beginning from 1), number of sampling points in this trace, sampling interval, shot ground elevation, and some other information. The discrete amplitude value, which is obtained from the sampled seismic signal after a certain time interval, is stored in trace data, and each sample point occupies 4 bytes (Nickersona et al., 1999).

(2) Data distribution. The distribution of SEG-Y data in a single file can be represented intuitively by frequency statistics; however, this method is too complicated to compare the data distribution among multiple files quantitatively and easily. Therefore, the Gini coefficient (G) (Nuti et al., 2015; Rey and Smith, 2013) was introduced to describe the distribution of SEG-Y files roughly in this paper. The formula for calculating G (Jianhua, 2007; Hoque and Clarke, 2015) is given as follows:

**Table 1**
SEG-Y file format.

| Volume header data (3200 bytes +400 bytes) | |
|---|---|
| Trace header data (240 bytes) | Trace data (Sampling points*4 bytes) |
| Trace header data (240 bytes) | Trace data (Sampling points*4 bytes) |
| ..... | ..... |
| Trace header data (240 bytes) | Trace data (Sampling points*4 bytes) |
| Trace header data (240 bytes) | Trace data (Sampling points*4 bytes) |

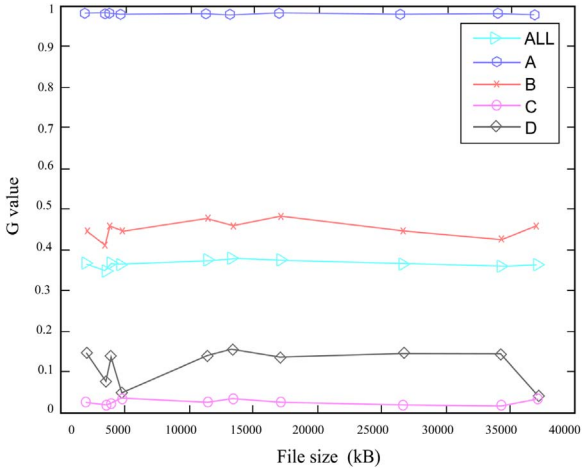


**Fig. 1.** Gini coefficient G of each data group with respect to SEG-Y file size.

$$G = 1 - \frac{1}{n}(2\sum_{i=0}^{n-1} W_i + 1), \quad (1)$$

where $n = 256$, $W_i$ is the percentage of the cumulative frequency of the 1st to the $i$th values (the frequency values are arranged in ascending order) of all cumulative frequencies. As G approaches a value of 1, the data distribution becomes increasingly uneven within the range [0, 255], and, in contrast, the data distribution becomes increasingly balanced as G approaches 0. To examine this distribution, 10 SEG-Y files of different sizes were selected, and the values of G for the overall data (no grouping) and for the data of each of groups A, B, C, and D (corresponding to the 4 bytes, where A as the first byte group, B as the second, and so on) of each file individually were calculated. The results are shown in Fig. 1.

As can be seen from Fig. 1, the G values calculated for each group were very similar regardless of the file size, which indicates that the distribution of data in each document within the range [0,255] is substantially similar. The different G values demonstrate that the data distributions of groups A, B, C, D, and the overall data are different from each other. Here, group A is observed to have a maximum G value close to 1, indicating that its data is distributed very unevenly within the range [0,255]. This is followed by group B, and, finally, by groups C and D, which represent considerably more balanced data distributions owing to their much smaller G values. The value of G for the overall data lies in the middle of the four subgroups, which represents an averaged result of the balanced and uneven distributions of the four data groups.

### 2.2. MiniSEED data format and distribution

The miniSEED data-packet is composed of a fixed header of 48 bytes, a variable header section, blockettes, and a data field (Mariotti and Utheim, 2006). As shown in Table 2, the record length in the data field is fixed at 4096 bytes (Canabrava, 2004; Augliera et al., 2011).

The data field contains a continuous record of the original data in a limited time frame from a given number of channels in a single station. The original data is stored in time domain form, which occupies mainly 32 bits (4 bytes) of memory in mainstream earthquake monitoring instruments at present. To improve the efficiency of data transmission