

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319313909>

# Data Compression Using Elias Delta Code

Article · August 2017

DOI: 10.23883/IJRTER.2017.3406.TEGS6

CITATIONS

9

READS

2,598

4 authors:



**Leni Marlina**

Universitas Pembangunan Panca Budi

9 PUBLICATIONS 42 CITATIONS

[SEE PROFILE](#)



**Andysah Putera Utama Siahaan**

Universitas Pembangunan Panca Budi

329 PUBLICATIONS 2,170 CITATIONS

[SEE PROFILE](#)



**Heri Kurniawan**

Universitas Pembangunan Panca Budi

2 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



**Indri Sulistianingsih**

Universitas Pembangunan Panca Budi

12 PUBLICATIONS 18 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Quality Assurance in Knowledge Data Warehouse [View project](#)



A comparative analysis of economic farming activity of farmers rice-livestock integrated farming system (RLIFS) and non RLIFS [View project](#)



## **Data Compression Using Elias Delta Code**

**Leni Marlina<sup>1</sup>, Andysah Putera Utama Siahaan<sup>2</sup>, Heri Kurniawan<sup>3</sup>, Indri Sulistianingsih<sup>4</sup>**

Faculty of Computer Science, Universitas Pembangunan Panca Budi, Medan, Indonesia

<sup>2</sup>Ph.D. Student of School of Computer and Communication Engineering, Universiti Malaysia Perlis, Kangar, Malaysia

**Abstract** — Compression is an activity performed to reduce its size into smaller than earlier. Compression is created since lack of adequate storage capacity. Data compression is also needed to speed up data transmission activity between computer networks. Compression has the different rule between speed and density. Compressed compression will take longer than compression that relies on speed. Elias Delta is one of the lossless compression techniques that can compress the characters. This compression is created based on the frequency of the character of a character on a document to be compressed. It works based on bit deductions on seven or eight bits. The most common characters will have the least number of bits, while the fewest characters will have the longest number of bits. The formation of character sets serves to eliminate double characters in the calculation of the number of each character as well as for the compression table storage. It has a good level of comparison between before and after compression. The speed of compression and decompression process possessed by this method is outstanding and fast.

**Keywords** — Compression, Elias Delta

### **I. INTRODUCTION**

The development of information at this time trigger the creators of storage media to develop technology on a harddisk, flash, CD, and others. It is due to the lack of capacity provided by certain media. As well as dual-screen DVDs that can only hold data of approximately 9 GB. Thus, created a new storage media such as blue ray disk that can store data larger than 30 GB. A common problem with data compression is how to optimally compress the data, especially text data that does not have a header file. It aims to obtain the compression results have a size that is much smaller than the original size. However, to make the compression process is not easy. Compression must be done correctly to the original information. If there is an error in the preparation of the bit sequence, then during the decompression process will fail. To support all that, it has to be seen the kind of compression that is suitable to apply to certain data.

Compression has two types, lossy and lossless techniques [2]. Lossy compression is a non-returnable compression to all forms. The discharges that occur in this compression can not return to the original data or in other words; this compression process has no decompression. Processes like this are often done to shrink the size of video, audio, images. This compression is done because of constrained storage media, especially on mobile phones or constrained in mobile phone resolution to display or listen to the media. The second technique is lossless compression. It is a common data compression for archiving where compression results can be returned to the decompression process to get the original data. The Elias Delta method is a lossless compression in the process of compacting the data. By applying this method to shrink the data, it is expected that the burden of storage media can be reduced.

## II. THEORIES

### 2.1 Compression

The rapid development of information requires that technology in all fields should flourish as well. It continually seeks to evolve to meet the needs of the community for information that can be quickly and easily accessed. To do this requires a large database. It deals with storage media. It also relates to data access process. If the data has a large capacity, it will slow down the access speed and reduce the capacity of the storage media quickly [7][8]. This problem makes developers of information technology perform data compression. The original data will be converted into smaller data but does not reduce the value contained in the information.

Compression is a technique of information reduction so that new data is obtained with a size smaller than the size before it is compressed [8]. It can be done on some types of data, video, images, audio, and others. The data in the image is a file with BMP, JPG, PNG, TIFF, BMP extension. Audio data is in the form of MP3, AAC, RMA, WMA and video in the form of MP4, WMV, AVI and so on. The compression process is the process of encoding information using bits or other information-bearing units lower than the data representation that is not encoded in the encoding system [3].

The compression technique has two ways:

- **Lossy Compression.** This technique is used to degrade the quality of the original media. Compressed data will remain equally visible or enjoyed like the original but with lower quality. It is more often used for streaming media, audio, images and video. The resulting file size will be smaller than the original but still be eligible to apply. This process works by removing unused portions of data. These data are not necessary for the sight of the human eye because it is not directly related to the user. So when the data has been compressed, the data can still be used even though it is not like the original. For example, a video that has a full HD 1920 x 1650 resolution. For this size, it will affect the speed at the time of video playback on the old computer, so that requires the reduction of resolution to HD 1280 x 720 or VGA 640 x 480.
- **Lossless Compression.** This compression is often done to save data storage or to combine some data into one data so as not to be scattered between one another. It is often done when sending emails with many attachments that minimize the attachment process. The compressed data can be returned to all forms. The result of the decompression process is no different from the original. An example can be seen in the WinRAR app. File compression results in the form of ZIP, RAR, GZIP, 7-Zip and so forth. The images, audio, video files can also be compressed with this method just like on lossy compression. However, the purpose is different, not to lower the quality or resolution, but to perform data archiving. The file size after the compression process is not necessarily smaller than the original file if the data is saved optimally.

### 2.2 Elias Delta

Peter Elias invented Elias Delta. This code applies methods such as Elias Gamma, especially in the head [1][4]. The technique is as follows:

- Find the highest binary rank, for example, 11 which has a binary value of 1011 where the highest rank is 3. So  $N' = 3$ .
- Use Gamma Coding to encode  $N$  numbers where  $N = N' + 1$ . So for the case of decimal 11 then we have to make a Gamma Coding of 4 that is 00100.
- Add the remaining  $N'$  binary in result no.2. So obtained answer 00100011.

- Next is the Elias Delta Coding method, the principle is the opposite of steps one through three above. Suppose we will decode 00100011.
- Find the number of zeros before it finds the number one, which is 00, amounts to two. Means there are  $(2 + 1)$  numbers to watch after these two zeros are 100 which in decimals mean 4, so we get N 'with  $N-1 = 4-1 = 3$ .
- If N 'is known, i.e. 3 then three remaining bits are part of that number i.e. 011. So obtained answer 1011 which means 11.

**Table 1. Elias Delta encoding system**

Number	N	N+1	Encoding	Probability
$1 = 2^0$	0	1	1	1/2
$2 = 2^1 + 0$	1	2	0 1 0 0	1/16
$3 = 2^1 + 1$	1	2	0 1 0 1	1/16
$4 = 2^2 + 0$	2	3	0 1 1 00	1/32
$5 = 2^2 + 1$	2	3	0 1 1 01	1/32
$6 = 2^2 + 2$	2	3	0 1 1 10	1/32
$7 = 2^2 + 3$	2	3	0 1 1 11	1/32
$8 = 2^3 + 0$	3	4	00 1 00 000	1/256
$9 = 2^3 + 1$	3	4	00 1 00 001	1/256
$10 = 2^3 + 2$	3	4	00 1 00 010	1/256
$11 = 2^3 + 3$	3	4	00 1 00 011	1/256
$12 = 2^3 + 4$	3	4	00 1 00 100	1/256
$13 = 2^3 + 5$	3	4	00 1 00 101	1/256
$14 = 2^3 + 6$	3	4	00 1 00 110	1/256
$15 = 2^3 + 7$	3	4	00 1 00 111	1/256
$16 = 2^4 + 0$	4	5	00 1 01 0000	1/512
$17 = 2^4 + 1$	4	5	00 1 01 0001	1/512
$18 = 2^4 + 2$	4	5	00 1 01 0010	1/512

Table 1 shows the encoding system of Elias Delta [5][6]. As an understanding, it can be seen an example, if  $n = 18$  produces the code delta 001010010. It is obtained from the following calculations:

- Take the closest the power of two,  $2^4 = 16$
- Add the power value 4 and 1 = 5 so that the number of characters in the first part is 00101
- The rest of the earlier power is  $18 - 16 = 2$
- Add the beginning part with bit value as much as four characters = 0010
- The result of the Elias Delta Code is 001010010

### III. RESULT AND DISCUSSION

At this stage, compression testing is performed for a simple string. It is "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG." Stages in this section include:

- Calculate the length of the text
- Create and calculate the length of the character set
- Calculating bit length
- Specifies padding bits if necessary

Table 2 shows the original string before the compression process. The string has 43 characters.

*Table 2. Original string*

No.	Char	ASCII	Binary
1	T	84	01010100
2	H	72	01001000
3	E	69	01000101
4		32	00100000
5	Q	81	01010001
6	U	85	01010101
7	I	73	01001001
8	C	67	01000011
9	K	75	01001011
10		32	00100000
11	B	66	01000010
12	R	82	01010010
13	O	79	01001111
14	W	87	01010111
15	N	78	01001110
16		32	00100000
17	F	70	01000110
18	O	79	01001111
19	X	88	01011000
20		32	00100000
21	J	74	01001010
22	U	85	01010101
23	M	77	01001101
24	P	80	01010000
25	S	83	01010011
26		32	00100000
27	O	79	01001111
28	V	86	01010110
29	E	69	01000101
30	R	82	01010010
31		32	00100000
32	T	84	01010100
33	H	72	01001000

34	E	69	01000101
35		32	00100000
36	L	76	01001100
37	A	65	01000001
38	Z	90	01011010
39	Y	89	01011001
40		32	00100000
41	D	68	01000100
42	O	79	01001111
43	G	71	01000111

The next stage is the formation of character sets. Repeated characters will be eliminated so that only single characters live. After this process, there are only 27 characters left with different frequency of occurrences. The number of bits in this string is 344. The result of the character set process can be seen in Table 3.

**Table 3. Elias Delta characters set**

No.	Char	ASCII	Binary	Freq.	Bits
1	T	84	01010100	2	16
2	H	72	01001000	2	16
3	E	69	01000101	3	24
4		32	00100000	8	64
5	Q	81	01010001	1	8
6	U	85	01010101	2	16
7	I	73	01001001	1	8
8	C	67	01000011	1	8
9	K	75	01001011	1	8
10	B	66	01000010	1	8
11	R	82	01010010	2	16
12	O	79	01001111	4	32
13	W	87	01010111	1	8
14	N	78	01001110	1	8
15	F	70	01000110	1	8
16	X	88	01011000	1	8
17	J	74	01001010	1	8
18	M	77	01001101	1	8
19	P	80	01010000	1	8
20	S	83	01010011	1	8
21	V	86	01010110	1	8
22	L	76	01001100	1	8
23	A	65	01000001	1	8
24	Z	90	01011010	1	8
25	Y	89	01011001	1	8
26	D	68	01000100	1	8
27	G	71	01000111	1	8
					<b>344</b>

According to the arrangement of bits of the characters in the Elias Delta table, the result of compression of the previous string is 248 as seen in Tabel 4. This result is obtained based on the use of bits according to the frequency of appearance of the characters. The space character is the one that most often appears in the string; There are eight occasions. There is a character that appears three and four times. Four characters appear twice and 20 characters each appearing just once.

**Table 4. Elias Delta result**

No.	Char	Freq.	Binary	Length	Bits
1		8	1	1	8
2	O	4	0100	4	16
3	E	3	0101	4	12
4	H	2	01100	5	10
5	U	2	01101	5	10
6	R	2	01110	5	10
7	T	2	01111	5	10
8	C	1	00100000	8	8
9	K	1	00100001	8	8
10	B	1	00100010	8	8
11	Q	1	00100011	8	8
12	I	1	00100100	8	8
13	W	1	00100101	8	8
14	N	1	00100110	8	8
15	F	1	00100111	8	8
16	X	1	001010000	9	9
17	J	1	001010001	9	9
18	M	1	001010010	9	9
19	P	1	001010011	9	9
20	S	1	001010100	9	9
21	V	1	001010101	9	9
22	L	1	001010110	9	9
23	A	1	001010111	9	9
24	Z	1	001011000	9	9
25	Y	1	001011001	9	9
26	D	1	001011010	9	9
27	G	1	001011011	9	9
					<b>248</b>

Bit Sequence:

011110110001011001000110110100100100001000000010000110010001001110010000100101001  
001101001001110100001010000100101000101101001010010001010011001010100101000010101  
010101011101011110110001011001010110001010111001011000001011001100101101001000010  
11011**00110000**

$$\begin{aligned}
 TB &= 248 \\
 TC &= \frac{TB}{8} \\
 &= \frac{248}{8} \\
 &= 31 \\
 \text{Padding} &= TB \bmod 8 \\
 &= 248 \bmod 8 \\
 &= 0 \\
 \text{Total Bits} &= 248 + 8 \\
 &= 256
 \end{aligned}$$

The bold character at the end of the bit sequence is the number of padding bits. For the above calculation, it is obtained that Padding = 0. Character "0" is on the order of 48 in ASCII code which after converted to binary will produce 00110000. This binary number will be added to the compression result, so the binary number becomes  $248 + 8 = 256$  bits. After the 256 bit is converted to a sequence of characters, this will generate the string {? F0B ????????? N ?? E ¥ "?? R UuYX®X, ÈH [0 as many as 32 characters.

$$\begin{aligned}
 \text{Compression Ratio} &= \frac{\text{after compression}}{\text{before compression}} * 100\% \\
 &= \frac{256}{344} * 100\% \\
 &= 74.41860465116279\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Redudancy} &= \frac{\text{before - after compression}}{\text{before compression}} * 100\% \\
 &= \frac{344-256}{344} * 100\% \\
 &= \frac{88}{344} * 100\% \\
 &= 25.58139534883721\%
 \end{aligned}$$

The compression process has saved data of 25.58139534883721% of the original data. The savings rate depends on the order and character pattern of the original message.

#### IV. CONCLUSION

The above calculations prove that the compression made by Elias Delta has been very good at saving storage capacity. The Elias Delta method can compress to all data types. The data will be converted into rows of character sets and then sorted by the frequency of appearance of characters. The most frequently displayed characters will vary by a document. The language also determines the number of most frequently used characters. Application of this compression is very well done to save storage media and speed up the process of sending data on a computer network. The downside of this compression is the decompression process. This compression requires additional tables to match the order of characters that often appear. Returning characters will take a long time because it must match the pieces of bits one by one as much as the data stored in the compression table.

#### REFERENCES

- [1] R. T. Handayanto, "Elias Gamma & Delta Coding," 15 September 2014. [Online]. Available: <https://rahmadya.com/2014/09/15/elias-gamma-delta-coding/>. [Diakses 24 August 2017].
- [2] S. D. Nasution dan Mesran, "Goldbach Codes Algorithm for Text Compression," *International Journal of Software & Hardware Research in Engineering*, vol. 4, no. 11, pp. 43-46, 2016.



- [3] Ameliachy, "Teknik Kompresi Data," 20 April 2012. [Online]. Available: <https://ameliachy.wordpress.com/2012/04/20/teknik-kompresi-data/>. [Diakses 24 August 2017].
- [4] Antoni, E. B. Nababan dan M. Zarlis, "Results Analysis of Text Data Compression on Elias Gamma Code, Elias Delta Code and Levenstein Code," *International Journal of Science and Advanced Technology*, vol. 4, no. 9, pp. 17-24, 2014.
- [5] Wikipedia, "Elias Delta Coding," [Online]. Available: [https://en.wikipedia.org/wiki/Elias\\_delta\\_coding](https://en.wikipedia.org/wiki/Elias_delta_coding).
- [6] P. Elias, "Universal Codeword Sets and Representations of the Integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, p. 194–203, 1975.
- [7] Suherman dan A. P. U. Siahaan, "Huffman Text Compression Technique," *SSRG International Journal of Computer Science and Engineering*, vol. 3, no. 8, pp. 103-108, 2016.
- [8] R. Aarthi, D. Muralidharan dan P. Swanminathan, "Double Compression of Test Data Using Huffman Code," *Journal of Theoretical and Applied Information Technology*, vol. 39, no. 2, pp. 104-113, 2012.