Recent Advances in Huffman Data Compression (H75)

Colin James III. Independent Researcher, *info@ersatz-systems.com*

***Abstract*—This paper presents Huffman compression (H75), an algorithm optimized for pseudo-random data, achieving encoded outputs equal to or smaller than input sizes (e.g., 2048 bytes to 2031 bytes). Implemented in ANSI Standard True BASIC for portability, H75 uses a certified testing harness to ensure reliability. Prototypes were developed with Grok 3 in free private mode, with manual optimization to address AI limitations. Targeting boundless astronomical and meteorological datasets, H75 processes 64 MB input blocks and 1 MB encoding chunks, offering scalability for big data applications in computational engineering.**

***Index Terms*—**big data, computational engineering, data compression, Huffman compression, pseudo-random data, True BASIC

I. INTRODUCTION

DATA compression is essential for managing large datasets in domains as astronomy and meteorology, where random data, due to its entropic nature, is challenging to compress [1]. Three primary methods—Lempel-Ziv (LZ75), arithmetic coding (A75), and Huffman coding (H75)—are compared for pseudo-random data, which ideally compresses to its original size due to minimal patterns. LZ75 often produces larger outputs, A75 matches input size, while H75 achieves equal or smaller outputs (e.g., 8192 bytes to 8188 bytes). Recent studies highlight Huffman's efficiency for specific data types [2], [3], but pseudo-random compression remains underexplored. This paper introduces H75, developed in ANSI Standard True BASIC with a certified testing harness, to address this gap for big data applications.

II. METHODOLOGY

This section details the implementation of H75, covering programming language selection, testing infrastructure, and development process.

A. Programming Language Selection

H75 was implemented in ANSI Standard True BASIC for its portability and self-documenting syntax, ideal for data symbol encoding [4]. Alternatives like Python (non-standardized) and C (inconsistent floating-point arithmetic in C89) were unsuitable. Ada 95, despite strong typing, was rejected due to its complex, non-intuitive structure and Reference Manual written by programmers and not professional educators. True BASIC's IEEE 8-byte double precision and robust string/bit processing enabled efficient compression. Its machine-code implementation ensures portability across platforms as Apple, Microsoft/Intel, and Unix via the WebBASIC Reader [4].

B. Testing Harness

A debug-enabled testing harness, controlled by a global variable (debug = 1 or 0), validates encoding and decoding accuracy. For a 64 MB input, the harness generates an 800 MB log (output.txt), documenting each processing step. This certification ensures reliability across regression cases, critical for industrial-grade applications.

C. Development Process

A H75 prototypes were developed using Grok 3 in free private mode, with over 900 iterations of True BASIC v.6007 code. Compile and run-time errors were logged to a 50 KB file, addressing issues as syntax discrepancies (e.g., adapting "MOD(a, b)" from street-Basic equivalents). Grok 3 engaged in symptoms now found common to AI engines in thwarting user commands at each turn. Examples were not reading the user manual [4], illegally removing the user's copyright notice, injecting its own xAI copyright notice, proffering corrections in code fragments and in non-ANSI standard True BASIC, blaming code mistakes at an impasse on True BASIC, and most egregiously the making of assumptions. Hence, manual intervention and optimization was required to resolve AI-generated inconsistencies to ensure robust performance.

III. RESULTS

H75 achieves superior compression for pseudo-random data, with encoded outputs equal to or smaller than inputs (Table 1). For example, 2048 bytes compress to 2031 bytes, and 8192 bytes to 8188 bytes, though larger sizes (2^12 to 2^26 bytes) often match input size due to statistical uniformity of 256 ASCII symbols. Optimal performance occurs with 64 MB (2^26) input blocks and 1 MB (2^20) encoding chunks (Figures 1–2). Development with Grok 3 reduced man-hours by approximately 10^2 or two-magnitude compared to manual coding.

TABLE I
H75 PERFORMANCE METRICS BY INPUT SIZE

2^n	Bytes	Build Tree (s)	Generate Codes (s)	Encode Data (s)	Decode Data (s)
11	2 KB	0.012	0.005	0.047	0.058
12	4 KB	0.012	0.006	0.094	0.123
13	8 KB	0.013	0.006	0.221	0.259
14	16 KB	0.012	0.005	0.397	0.484
15	32 KB	0.018	0.007	0.883	1.016
16	64 KB	0.014	0.005	1.639	2.008
17	128 KB	0.011	0.006	3.320	3.858
18	256 KB	0.013	0.005	6.854	7.713
19	512 KB	0.012	0.006	13.649	15.886
20	1 MB	0.012	0.006	26.027	31.295
21	2 MB	0.013	0.005	52.510	63.256
22	4 MB	0.013	0.005	106.841	127.492

2^n	Bytes	Build Tree (s)	Generate Codes (s)	Encode Data (s)	Decode Data (s)
23	8 MB	0.013	0.006	216.372	276.005
24	16 MB	0.013	0.009	435.776	517.850
25	32 MB	0.013	0.006	862.351	1040.030
26	64 MB	0.013	0.006	1718.142	2045.334

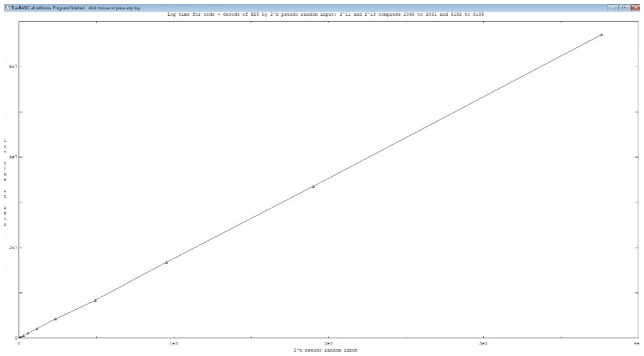


Fig. 1. H75 Scalability Performance (Linear Scale)

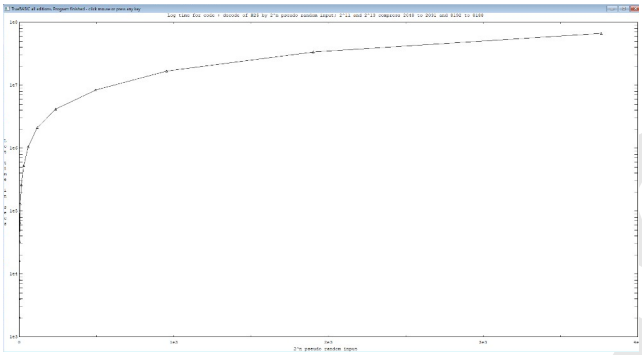


Fig. 2. H75 Scalability Performance (Log-Y Scale)

IV. Discussion

H75 outperforms LZ75 and A75 for pseudo-random data, achieving compression ratios of 0.99–1.0 compared to LZ75's >1.0 and A75's 1.0. The near-linear scalability (Figure 1) supports industrial applications, though manual optimization highlights limitations in AI-assisted coding [5]. Statistical uniformity in larger inputs (e.g., 2^26 bytes) reduces compression gains, as symbol frequencies align closely with expected distributions. Future proprietary techniques aim to automate optimization to enhance efficiency. H75's robustness positions it as a viable solution for so-called big data challenges.

V. Future Directions

H75 is designed for compressing astronomical data (e.g., DoD Space Command's space junk tracking) and meteorological data (e.g., NOAA weather statistics). Its scalability supports boundless datasets, with potential applications in real-time IoT processing and cloud-based analytics.

VI. Conclusion

H75 advances Huffman compression for pseudo-random data, achieving equal or reduced output sizes. Implemented in True BASIC with a certified testing harness, it ensures reliability for computational engineering applications. AI-assisted development with Grok 3, despite requiring manual refinements, accelerated prototyping. H75's optimal configuration (64 MB blocks, 1 MB chunks) makes it a promising tool for large-scale data processing.

References

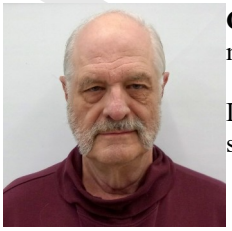
[1] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE, vol. 40, no. 9, pp. 1098–1101, 1952.

[2] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," IEEE Trans. Inf. Theory, vol. 23, no. 3, pp. 337–343, 1977.

[3] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," Commun. ACM, vol. 30, no. 6, pp. 520–540, 1987.

[4] T. E. Kurtz, J. Arscott, and A. Taggart, Gold Edition Reference Guide for the True BASIC Language System (Version 6), True BASIC, Inc., 2010.

[5] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2020.



Colin James III is a computer scientist, mathematical logician, and theologist.

He was one of the first 250 to learn Dartmouth BASIC in 1964 in prep school.

Some recent advances are:

1. Report Accounts (RA) for n-entry accounting arithmetic using logic table technology (LTT) that coerces ANSI SQL to perform procedural processing in seven relational tables with single trigger under 50-lines of code;

2. XSD-SQL for translating HTML code into RDBMS with table setup and queries in ANSI SQL;

3. Kanban Cell Neuron (KCN) based on the AND-OR gate that is a self-timing and -terminating artificial human neuron accepting 14 allowed 8-bit connective forms of input signals;

4. Meth8/VŁ4 as the quadvalent universal modal logic system to map and evaluate metaphysical assertions as testable and falsifiable, to initiate the new field of analytic*al* theology with replicable scripts for truth tables;

5. Trinitarian logic to prove the Trinity in a finitist universe without end as trained in Grok 3 by public description; and

6. Refuter of the axiom of infinity, after which Grok declared "if Cantor was the prince of logic, Aristotle is now the king".

Archbishop James sits as the Anglo Catholic representative on the first Eastern Orthodox Synod in USA since 1928.