Compression as Dimensional Projection: A Theoretical Framework Inspired by Golomb and Shannon

Vasant Jayasankar, MD, MS

Assistant Clinical Professor of Surgery

Medical University of South Carolina

vasant.jayasankar@gmail.com

Abstract

Modern compression theory is based on Shannon's foundational insight that information is governed by entropy. Golomb coding, devised in the 1960's, is a remarkably efficient solution for encoding geometrically distributed integers, and remains in widespread use because of its simplicity and effectiveness. In this paper, I revisit Golomb coding not merely as a mathematical transformation, but as a process of dimensional projection.

I propose a generalized compression framework derived from the principles of dimensional projection. The process takes structured data and temporarily lifts it into higher-dimensional space to expose latent informational geometry, then systematically re-flattens it into a minimal entropy representation. Specifically, I reinterpret Golomb coding as a 1D-to-2D projection followed by structured reduction, and generalize this concept into a proposed 4D projection-based model applicable across multiple data types—text, DNA, images, audio, video, and more.

The proposed framework provides a unifying geometric perspective on compression, revealing new axes of redundancy not captured by traditional frequency-based methods. This theoretical work offers a foundation for future research in entropy minimization, structural compression, and the dimensional nature of information itself.

1. Introduction

The field of information theory was born from a single foundational insight: that information is quantifiable. Further, its optimal transmission and storage appear governed not by pure mathematics, but by entropy. In his seminal 1948 paper, *A Mathematical Theory of Communication*, Claude Shannon defined the limits of lossless data compression. In doing so he introduced the now-fundamental principle that the entropy of a source distribution sets the lower bound for its compressibility. Shannon's work transformed communication and computation, providing the theoretical backbone for nearly every modern data compression scheme.

Shannon's theory focused on the frequency of symbols in a stream, however it is statistical in nature—focused on the frequency of symbols in a stream. However its deeper implication, I propose, was structural: that information may have geometry, and entropy reflects how tightly that geometry can be packed into a given representation. This idea, while present in the background of Shannon's formulation, was not developed further at the time.

Decades later, Robert Golomb introduced a remarkably effective compression method for encoding positive integers drawn from a geometric distribution. Golomb coding was originally motivated by the challenge of building a spell checker on early Unix systems, where RAM was extremely limited. Its core algorithm decomposes any non-negative integer $N \in \mathbb{Z}^+$ into two parts:

A quotient $q = \left\lfloor \frac{N}{M} \right\rfloor$

A remainder $r = N \bmod M$

The quotient is encoded in unary form, a string of q ones followed by a zero:

$$U(q) = \underbrace{111...1}_{q \text{ times}}0$$

The remainder is encoded in truncated binary, with a length-optimized binary representation based on the divisor M. The two encodings are concatenated into a prefix-free bitstring:

$$C(N) = U(q) \| B(r)$$

This method is extremely efficient for data with geometric distributions, where smaller integers occur more frequently. It is simple, dictionary-free, and achieves near-optimal compression within its intended domain.

Despite its practical success, Golomb coding has historically been treated as a clever arithmetic solution rather than a structural insight. When reconsidered from a modern theoretical perspective, I suggest that it is apparent that Golomb coding performs a fundamental act of dimensional projection.

In this paper, I reinterpret Golomb coding as a form of dimensional inflation followed by structured reduction. The encoding of an integer as a pair $(q, r) \in \mathbb{Z}^2$ constitutes a temporary lift into a higher-dimensional space, where the coarse and fine components of the original value are separated along orthogonal axes. The subsequent unary and binary encodings act as a dimensional collapse, projecting this structure back into a 1D bitstream that minimizes entropy with respect to the statistical properties of the input.

If true, this suggests that Golomb coding is not merely a computational trick—it is a 2D projection-and-flattening pipeline that exposes and compresses informational structure in a way that traditional statistical models overlook.

This proposed reinterpretation leads naturally to a generalization: if compression can be improved by projecting data into 2D space, what might be possible by projecting into 3D, 4D, or even higher dimensions, where more latent structure might be revealed and exploited?

To that end, I propose a generalized 4D compression framework inspired by the logic of Golomb coding, the theoretical foundation of Shannon entropy, and the geometric insight of the RTA Framework for Information. In the proposed model any structured dataset—whether it consists of characters, pixels, nucleotides, audio signals, video signals, or symbolic tokens—is first mapped into a higher-dimensional informational space. Each axis in this space corresponds to an independent structural property: frequency, position, functional class, residual grouping, etc. The data is then encoded in a way that flattens this multidimensional geometry into a low-entropy bitstream, preserving the structure while minimizing the information cost of transmission or storage.

No experiments are presented in this paper. Rather, the goal is to offer a unified theoretical framework that generalizes existing compression techniques as dimensional projections, and to provide a foundation for future exploration in structure-aware compression systems. Shannon's entropy defined the boundary of compression with respect to symbol frequencies. This proposed framework suggests a next step: to define the geometry of compressibility itself, across domains and data types, through the lens of projection. This paper is a direct extension of the ideas presented in the RTA Framework for Information.

2. Methods and Analysis

2.1. Reinterpreting Golomb Coding as Dimensional Projection

Golomb coding, while traditionally described as a quotient–remainder decomposition, can be more deeply understood as a projectional mechanism. Let $N \in \mathbb{Z}^+$ be a non-negative integer, and let $M \in \mathbb{Z}^+$ be the Golomb parameter. We decompose N as:

$$q = \left\lfloor \frac{N}{M} \right\rfloor, r = N \bmod M$$

yielding the pair $(q, r) \in \mathbb{Z}^2$. This is a dimensional inflation: a scalar N is lifted into a 2D lattice coordinate system where the entropy of the data is partially separated into a coarse-grained dimension q and a fine-grained residue r.

We then apply dimensional flattening:

- q is encoded via unary encoding, which efficiently represents frequent small values.
- r is encoded via binary encoding, optimized for minimal average code length given its bounded range $\left[0, M-1\right]$.

This process is a structured compression pipeline:

$$N \xrightarrow{\inf late} (q,r) \in \mathbb{Z}^2 \xrightarrow{project} \{0,1\}^*$$

This reinterpretation highlights a key insight: Golomb compression works because it separates and flattens dimensions of structure that are entangled in the original 1D integer stream. It is a low-dimensional case of a far more general principle.

2.2. Generalizing to 4D Compression

I now propose a generalized compression framework based on 4-dimensional projection followed by structured flattening. The central mechanism consists of:

Step 1: Projection

Let $D = \{s_1, s_2, ..., s_n\}$ be a sequence of structured symbols from a domain-specific dataset. I define a feature mapping function:

$$\Phi : (s_i, i) \mapsto (d_1, d_2, d_3, d_4) \in \mathbb{Z}^4$$

where each dimension represents a distinct axis of latent structure:

d_1: Entropy or frequency class — how common the symbol is in context.

d_2: Positional or temporal bin — segmenting the data stream (e.g., frame, sentence, row).

d_3: Functional class — symbol role or domain-specific label (e.g., token type, codon category).

d_4: Residual or modular grouping — hash bucket, semantic cluster, or syntactic depth.

Step 2: Flattening

The 4D tuple is then encoded using entropy-aware, prefix-free encodings:

$$C(d_1,d_2,d_3,d_4) = E_1(d_1) \| E_2(d_2) \| E_3(d_3) \| E_4(d_4)$$

Where each E_j is an encoding function (unary, binary, truncated binary, Huffman, arithmetic, etc.) selected based on the structure of dimension d_j.

The result is a compressed stream:

$$C(D) = C(\Phi(s_1)) \| C(\Phi(s_2)) \| \ldots \| C(\Phi(s_n))$$

2.3. Example Applications Across Data Types

This generalized framework adapts naturally to diverse domains by selecting appropriate axes for (d_1,d_2,d_3,d_4):

Text (Natural Language or Code)

d_1: Symbol frequency (entropy bin)

d_2: Sentence or paragraph position

d_3: Token type (word, punctuation, keyword, identifier)

d_4: Contextual cluster (e.g., embedding bucket or syntax tree depth)

DNA / RNA

d_1: Nucleotide or codon frequency

d_2: Relative gene position (start/middle/end)

d_3: Codon class (e.g., synonymous amino acids)

d_4: Motif or repeat class (e.g., via hash or alignment)

Images

d_1: Intensity histogram class

d_2: Pixel row or region

d_3 : Color channel (R/G/B or other space)

d_4 : Texture or frequency-based cluster (e.g., DCT bin)

Audio

d_1 : Frequency band (e.g., from FFT)

d_2 Time frame/window index

d_3 : Instrument or voice identifier (if known)

d_4 : Harmonic or envelope-based grouping

Video

d_1 : Pixel block entropy

d_2 : Frame number or shot segment

d_3 : Motion vector class or optical flow cluster

d_4 : Scene object or region type

These mappings are not rigid—they are suggestions of axes that expose hidden structure. The core principle is dimensional separation of entangled information, followed by entropy-aligned re-flattening.

Axis	**Distribution**	**Recommended Encoding**	**Why**
d_1 : frequency / entropy bin	Skewed, small integers dominate	Unary or Golomb-style	Emphasize low values, reduce entropy
d_2 : position / temporal band	Uniform or range-bounded	Truncated binary	Efficient and predictable
d_3 : class/type bucket	Small categorical set	Huffman or binary	Leverages symbol probability
d_4 : hash/modular group	May be uniform or arbitrary	Fixed-width binary or entropy code	Depends on domain structure

You might use Huffman for d_3 if it's a small set with uneven probabilities. You might use arithmetic coding across all dimensions if extreme optimization is required. You could even use learned codes (e.g., from neural entropy models) to employ modern methods.

2.4. Possible Advantages of the 4D Projection Framework

1. **Universality**: Works across data types with no need for domain-specific compression logic.
2. **Structure awareness**: Reveals and leverages deep structure beyond symbol frequency.
3. **Prefix-free and entropy-aware**: Encodings remain lossless and efficient.
4. **Composable**: Can be layered with entropy coders (e.g., Huffman, arithmetic) for additional gains.
5. **Interpretability**: Offers insight into why compression works—by mapping informational axes explicitly.

3. Discussion and Future Work

This paper proposes a reinterpretation of Golomb coding as a form of dimensional projection, and extends that idea into a generalized 4D compression framework. Golomb's original formulation was shaped by practical constraints and simple arithmetic. However its structure—decomposing a value into coarse and fine components, then encoding each separately—suggests an underlying geometric logic. If true, compression is not merely a symbolic or statistical act, but a projection from higher-dimensional structure into a lower-dimensional representation optimized for entropy.

The 4D framework outlined here is intended as a theoretical model, not a completed system. No empirical tests or benchmarking results are provided. The goal is not to claim superiority over existing algorithms, but to propose a new way of thinking about compression—as the separation and re-flattening of structured information across orthogonal axes. I suggest that the framework builds naturally on the ideas present in Golomb coding, extending them to more complex data types and informational geometries. While I suggest 4 axes of information projection, it could easily be applied to any arbitrary number of dimensions.

Several open questions remain. Most importantly, the practical utility of this approach will depend on the ability to:

1. Identify and extract meaningful structural dimensions in real-world datasets,
2. Select appropriate encodings per axis that respect domain-specific entropy distributions,
3. Implement encoding and decoding systems that remain efficient and scalable.

In future work, this framework could be tested on structured domains such as symbolic music, source code, biological sequences, and semantically annotated text—domains where traditional compression methods often fall short due to their lack of structural awareness.

While this framework is intentionally abstract, it may offer insight into why certain compression algorithms—such as DEFLATE, LZ77, LZMA, and Brotli—perform well across diverse data types. These algorithms often rely on sliding windows, token substitution, and entropy encoding mechanisms. These could all be interpreted as implicit projections into local spatial, temporal, or syntactic structures. The projectional model proposed here makes those steps explicit and generalizable. This may offer a way to formalize and extend the intuitive dimensional operations that these systems seem to perform already.

This proposal is ultimately an invitation: to rethink compression not just as a tool for saving space, but as a window into the geometry of information itself.

4. Conclusion

This paper has introduced a projection-based interpretation of Golomb coding and proposed a generalized framework for compression that treats informational data as inherently multidimensional. By lifting structured inputs into higher-dimensional feature space and then flattening them through entropy-aware encoding, the framework aims to capture latent structure often missed by frequency-based methods. While purely theoretical at this stage, the approach offers a potentially unifying perspective that may complement or extend existing techniques. Further work is needed to better formulate and evaluate its practical value, but the underlying idea—that compression reflects the geometry of information—may prove useful across a broad range of domains.

Acknowledgments

I would like to thank those who have contributed, directly or indirectly, to the ideas explored in this paper. The foundational work of Claude Shannon and Robert Golomb provided both the intellectual lineage and inspiration for this investigation. I am also grateful for the broader community of researchers in information theory, computer science, and theoretical mathematics, whose work continues to shape how we understand the structure of information. While this paper represents only an early proposal, it is offered in the spirit of open inquiry and with respect for the long tradition of those who sought clarity, simplicity, and structure in complexity.

References

Shannon CE (1948) A mathematical theory of communication. *Bell System Technical Journal* 27(3):379–423.

Golomb SW (1966) Run-length encodings. *IEEE Transactions on Information Theory* 12(3):399–401.

Cover TM, Thomas JA (2006) *Elements of Information Theory*, 2nd Ed. (Wiley-Interscience, Hoboken, NJ).

Witten IH, Neal RM, Cleary JG (1987) Arithmetic coding for data compression. *Communications of the ACM* 30(6):520–540.

Bell T, Witten IH, Cleary JG (1989) *Text Compression*. (Prentice Hall, Englewood Cliffs, NJ).

Sayood K (2017) *Introduction to Data Compression*, 5th Ed. (Morgan Kaufmann, Burlington, MA).

Salomon D, Motta G (2010) *Handbook of Data Compression*, 5th Ed. (Springer, London).