Research Overview

I'm conducting research on an algorithm utilizing a NxNxN 3D data structure akin to a Rubik's cube. This structure uses data bits (0s and 1s) instead of colors on its faces. The goal is to maximize the information encoded on the six faces with minimal twisting movements.

Key Points

1. **Data Structure**: The NxNxN cube has six faces, each with NxN slots, totaling NxNx6 data slots. Not all slots are usable due to symmetry and fixed positions.
2. **Compact Twisting Movements**: We need a compact descriptor for twisting movements that requires less data than the information it encrypts.

Detailed Analysis

Usable Data Slots Calculation

- **Total Data Slots (ds)**: Each of the six faces has $N \times N$ slots.
- **Example Calculation (N = 4)**:
  - Total slots: $6 \times 16 = 96$ ds.
  - Usable slots (Z): Some slots are fixed due to symmetry. For instance, center and corner slots may be fixed by symmetry.
  - Usable variable ds: $Z = 42$ (assuming 54 fixed bits based on a specific encoding scheme).

Twisting Movement Descriptor (T)

- **Generic Descriptor for N ranging from 3 to 7**:
  - Use 3 bits to address rows and columns (0 to 7):
    - Values 0-3: Outer slices
    - Values 4-7: Inner slices
    - Value 0: No rotation
    - Value 7: Two-layer rotation
  - Use another 2 bits for rotation direction:
    - `00`: No rotation
    - `01`: Clockwise
    - `10`: Counterclockwise
    - `11`: Half-turn
- Total bits per move (T): 6 bits.

Solving Initial Configuration (I)

- **Example Calculation for N = 4**:
  - **Corners**: $(2^8) \times 8!$ possibilities.
  - **Edges**: $(2^{24}) \times 24!$ possibilities.
  - **Centers**: 2^{24} possibilities.
  - Total possibilities (I):

$$\frac{(2^8) \times 8! \times (2^{24}) \times 24! \times (2^{24})}{24}$$

  - Logarithm base 2:

$$\log_2\left(\frac{(2^8) \times 8! \times (2^{24}) \times 24! \times (2^{24})}{24}\right) \approx 197.9$$

Information Payload (P)

The maximum information payload P is related to I and Z. If each configuration I can be mapped to a set of Z bits, the payload P in bits per encoding/decoding cycle is:

$$P = \log_2(I) \approx 146.74 \text{ bits (for N = 4)}$$

Minimum Viable Cube Size Equation

To ensure efficient encoding, the cube size N must satisfy:

$$\frac{2N + \log_2(N!)}{6N^2 - kN} \geq 1$$

This inequality can be solved numerically to estimate the minimum viable cube size N.

Conclusion

The theoretical framework and calculations suggest that the approach is potentially viable. For $N = 4$, the efficiency is approximately 3.49 bits per data slot. Further optimization and testing on larger cube sizes, as well as practical implementation, are necessary to validate and enhance the algorithm's efficiency and scalability.