

# 視覚野刺激型人工視覚システムの重み付き刺激データ表現の符号化手法の提案

上田 匠† 武内 良典‡  
Takumi Ueda Yoshinori Takeuchi

## 1. はじめに

本論文では、シリコン網膜を搭載した視覚野刺激型人工視覚システムの実現に向けた刺激データの符号化方法の提案をする。シリコン網膜とは、アナログ CMOS VLSI によって生体の視覚系の回路構造および機能を実現したものを指し、リアルタイム性を満足するため高速な画像処理能力をシリコン網膜が持つ超並列アナログ知能知覚センサ [1] で実現されている。

シリコン網膜では、システムを使用する人間は電極を制御するユニットを人体の頭蓋骨内部に埋め込んだ状態で装着する。また、システムでは体外と体内のシステムの間を無線通信により情報を送受信する。システムの電力消費によって生じる熱が人体へ影響を及ぼさないようにするためには、システムの消費電力量を小さくする必要がある。体内と体外間のデバイスの無線通信が人工視覚システム内で消費する電力がシステム中の大きな割合を占めると見積もられるため、送信データに対して符号化を用いることにより、情報圧縮処理を行ってデータ量を削減し無線通信のビットレートを下げることで消費電力量を抑えることがシステム全体の消費電力量削減に対して有効である。

本研究では、シリコン網膜を用いた視覚野刺激型人工視覚システムが発する熱量による人体への影響を減らすために、高いデータ削減率と低消費電力量を兼ね揃えた新たな情報圧縮を行うための符号化方法を検討する。

## 2. 視覚野刺激型人工視覚システム

### 2.1 システムの概要

視覚野刺激型人工視覚システムは、シリコン網膜システム、データ処理や誤り訂正符号化する体外プロセッサ、無線通信のユニットで構成される。シリコン網膜とは、アナログ CMOS 集積回路を用いて、超並列画像処理機能を実現することで網膜の処理を実装したアナログビジョンチップシステムである [1]。これにより網膜の機能を模した高速な画像処理が可能となる。体外プロセッサではデータバッファの役割とともに、データを誤り訂正符号化し体内側のシステムへ送信する。

### 2.2 データ処理の流れ

シリコン網膜を用いた視覚野刺激型人工視覚システムのデータの流れを図 1 に示す。

1. シリコン網膜が外界の画像を入力として取り込む。
2. シリコン網膜から視覚野刺激データが出力され、視覚野刺激データを体外プロセッサで誤り訂正符号化する。
3. 体外プロセッサから無線通信ユニットへ誤り訂正符号化された視覚野刺激データが送られ、無線通信に

よって体外から体内部分の無線通信ユニットへ送信される。

4. 体内部分の無線通信ユニットが体内プロセッサへ誤り訂正符号化された視覚野刺激データを送信し、体内プロセッサで誤り訂正を行う。
5. 体内プロセッサが視覚野刺激デバイスのバッファレジスタへ視覚野刺激データを送信する。
6. バッファレジスタに格納された視覚野刺激データに沿って視覚野刺激デバイスの多極電極を操作し、視覚野へ電氣的刺激を与える。
7. 与えられた刺激により、システムを装着した人間が phosphene を感じる。

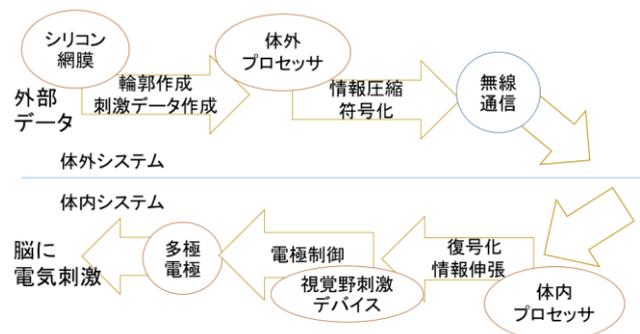


図 1 シリコン網膜のデータ処理の流れ

### 2.3 システムの要求仕様

一般的に無線通信による電力消費がシステム全体で大きな割合を占める。無線通信量が大きいと、無線通信による電力消費が大きくなり、システム全体が発する熱量が増加して内部と外部のユニット間に存在する生体組織の温度が上昇し、身体に悪影響を及ぼすことが懸念される [2]。更に、人体の視覚以外の感覚との認識のずれを防ぐためシステムの動作と他の感覚器官の感覚の遅延を少なくする必要もある。すなわち、本システムは、低消費電力量、リアルタイム性、高い情報圧縮性の実現が必要である。

## 3. ランレングス符号化とエイリアス符号

### 3.1 ランレングス符号化 [3]

0 と 1 の数字で構成する符号の中で、0 又は 1 の連なりを一つのブロックとし、このブロックに長さを表す符号を割り当てる。ランレングス符号化は単純な画像データほど効果的な圧縮が可能であり、可逆圧縮であるため完全にデータを再現できるというメリットを持っている。ランレングス符号化の例を表 1 に示す。

† 近畿大学大学院総合理工学研究科, Graduate School of Science and Technology, Kindai University

‡ 近畿大学理工学部, Faculty of Engineering, Kindai University

表1 ランレングス符号化の例

情報ビット	ランレングス符号化後の 数値データ
0101	01 11 01 11
0000	04
1111	14
00001111	04 14

表1のランレングス符号化の例を説明する。ランレングス符号化は、データの連なりを一つのブロックとする。情報ビットが0101の場合、0、1、0、1の情報がそれぞれ1回ずつ出現している。したがって、0が1連続であることを表す01、1が1連続であることを表す11、同様に01、11を用いて、0101の情報ビットは01110111と符号化される。0000のように、情報ビット0が4連続の場合は、04のように符号化される。

### 3.2 エイリアス符号[4]

#### 3.2.1 $\gamma$ 符号化

$\gamma$ 符号化は入力データに対し、二進数で表現された入力データのビット幅から1引いた値の0を出力することで接頭語とし、入力データの二進数表現のビットを出力する。 $\gamma$ 符号化の例を表2に示す。

表2  $\gamma$ 符号化の例

整数値	ビット幅 (2進数)	符号語
1	1(1)	1
2	2(10)	010
3	2(11)	011
4	3(100)	00100
5	3(101)	00101
...	...	...

表2の $\gamma$ 符号化の例を説明する。整数値1は二進数で(1)であり、ビット幅は1である。 $\gamma$ 符号化のヘッダは、ビット幅の値から1引いた値の0を出力するのでヘッダは出力せず、入力データの二進数表現である1を出力する。整数値1の $\gamma$ 符号化は1と表現される。

整数値2は二進数で(10)であり、ビット幅は2である。したがって、 $\gamma$ 符号化のヘッダは、ビット幅の値から1引いた値の0を出力するので0をヘッダとし、入力データの二進数表現である10を出力する。整数値2の $\gamma$ 符号化は010と表現される。

#### 3.2.2 $\delta$ 符号化

$\delta$ 符号化は入力データに対し、二進数で表現された入力データのビット幅を $\gamma$ 符号として出力することで接頭語とし、入力データの二進数表現のビットから最上位ビットを省いたものを出力する。 $\delta$ 符号化の例を表3に示す。

表3  $\delta$ 符号化の例

整数値	ビット幅	ビット幅の $\gamma$ 符号化	符号語
1	1	1	1
2	2	010	0100
3	2	010	0101
4	3	011	01100
5	3	011	01101
...	...	...	...

表3の $\delta$ 符号化の例を説明する。整数値1は二進数で(1)であり、ビット幅は1である。ビット幅に $\gamma$ 符号化を用いると表2より符号語は1であり、 $\delta$ 符号化のヘッダは1となる。整数値1は二進数で(1)であり、最上位ビットを省く。整数値1の $\delta$ 符号化は1と表現される。

整数値2は二進数で(10)であり、ビット幅は2である。ビット幅に $\gamma$ 符号化を用いると表2より符号語は010であり、 $\delta$ 符号化のヘッダは010となる。整数値2は二進数で(10)であり、最上位ビットを省いた0とヘッダを接続する。整数値2の $\delta$ 符号化は0100と表現される。

### 3.3 $\gamma$ 符号化と $\delta$ 符号化の比較

16以下の整数値に対して $\gamma$ 符号化と $\delta$ 符号化を用いた時のビット数の比較を図2に示す。1024以下の整数値に対して $\gamma$ 符号化と $\delta$ 符号化を用いた時のビット数の比較を図3に示す。

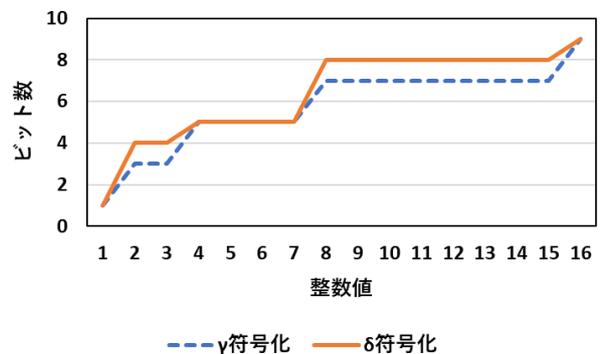


図2 16以下の整数値に対して $\gamma$ 符号化と $\delta$ 符号化を用いた時のビット数の比較

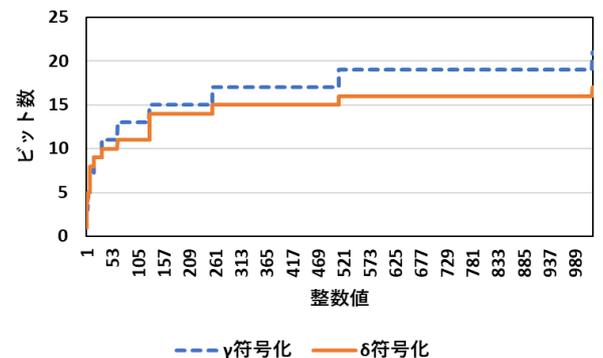


図3 1024以下の整数値に対して $\gamma$ 符号化と $\delta$ 符号化を用いた時のビット数の比較

図2、図3は横軸が整数値で、縦軸が整数値を符号化した時のビット数である。本研究では、少ないビット数で表現したほうが効果的で、横軸の整数値に対して縦軸のビット数が少ない符号化が優れていることを示す。

図2から、16以下の整数値は $\gamma$ 符号による符号化は、 $\delta$ 符号による符号化よりも少ないビット数で値を表現できることがわかる。また、図3から、48以上の整数値は、 $\delta$ 符号による符号化は、 $\gamma$ 符号による符号化よりも少ないビット数で値を表現できることがわかる。

## 4. 視覚野刺激データ符号化

### 4.1 視覚野刺激データの構成

本研究で扱う視覚野刺激データは、128×128の2次元配列状に与えられる。電極1つを1点で表し、視覚野に設置されたどの刺激電極を駆動するかで表現する。1つ1つのデータは信号の有無を表すため、計算機上ではバイナリデータとして表されている。0ならばLowで視覚野に刺激を送らず、1ならばHighで視覚野に刺激を送ることを表している。視覚野に刺激を送ると光の点(phosphene)を脳が認識し、疑似的な視覚を得ることができる。

### 4.2 フレーム加算による重み付き刺激データ

従来の手法[2]では、視覚野刺激データを1フレームごとに差分を取って処理を行っていた。しかし、刺激データ1フレームに出現する刺激点(データの値が1)の発生確率は、約0.5%程度と非常に少ない。そのことから、フレームを1枚ごとに処理するのではなく、フレームを一定の枚数加算することで、まとめて処理することを検討した。具体的には、視覚野刺激データを50フレームから150フレームずつ加算して、重み付き刺激を作成した。重み付き刺激の例を図4に示す。

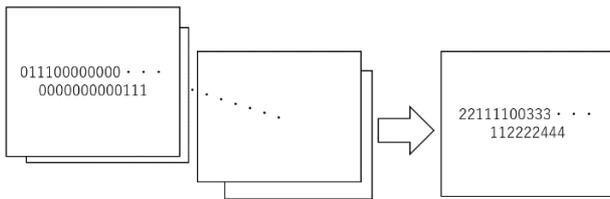


図4 フレーム加算による作成した重み付き刺激データの生成

図4のフレーム加算後の刺激データの大きさは、一定の枚数加算したフレーム内に刺激データが同じ位置に存在する回数を示す。刺激データの値が2の時は、一定の枚数加算したフレーム内に刺激データ1が2回存在することを表す。

### 4-3 重み付き刺激データの特徴

今回使用したサンプルは約7秒で14000フレームである。人が認識できるフレーム数には十分大きなフレーム数と考えられるので、重み付き刺激データを作成することにより、データ量を減らすことができると考えられる。また、重みを利用して刺激の強さを変更することにより、視覚認知を高めることもできると考えられる。

符号化を用いてデータを圧縮するには、符号化をするデータの特徴を調べる必要がある。そのため、50フレームから150フレーム加算することにより、作成した重み付き刺激データの加算した値の出現率を調べた。図5に50フレーム加算した時の重み付き刺激データの値の出現率、図6に100フレーム加算した時の重み付き刺激データの値の出現率、図7に150フレーム加算した時の重み付き刺激データの値の出現率を示す。

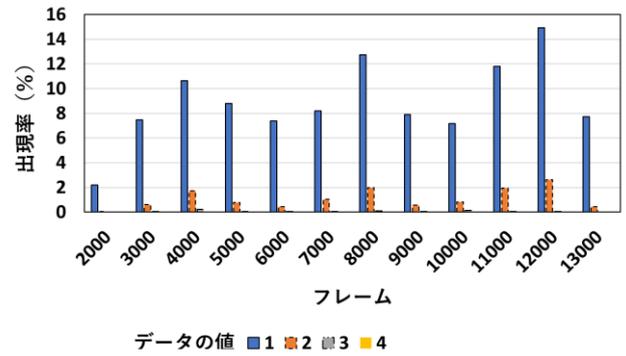


図5 50フレーム加算した時の重み付き刺激データの値の出現率

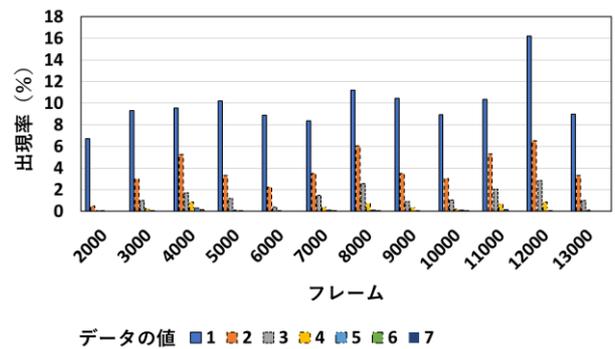


図6 100フレーム加算した時の重み付き刺激データの値の出現率

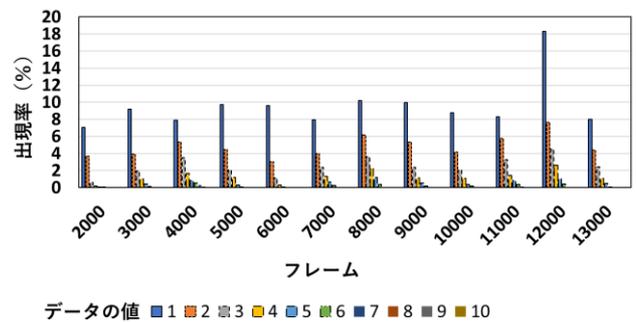


図7 150フレーム加算した時の重み付き刺激データの値の出現率

図5、図6、図7の横軸は刺激データの14000フレーム中のフレーム番号を示し、縦軸は刺激データの値の出現率を示す。1000フレーム目や14000フレーム目は、刺激データの値が全て0である。フレームの加算枚数により、重み付き刺激データの値の出現率の変化がわかる。フレームの加算枚数を増やしても、重み付き刺激データの出現する値の多くが1や2などの小さな値である。そして、刺激データの値が大きくなるほど、データの出現率が小さくなることがわかる。すなわち、整数値1や2に対して有効な符号化が重み付き刺激データの符号化に有効であるとわかる。

#### 4-4 提案手法

視覚野刺激データに対する提案手法の流れを示す。

1. 視覚野刺激データのフレームを加算して重み付き刺激データを作成する。
2. 重み付き刺激データに対してランレングス符号化を用いて整数値データを作成する
3. 作成した整数値データに対して算術符号化である  $\gamma$  符号化、 $\delta$  符号化を用いる
4.  $\gamma$  符号化、 $\delta$  符号化を用いて得られたビット数から圧縮率を比較する

#### 5. 視覚野刺激データ符号化後の圧縮率比較

##### 5-1 圧縮率の定義

圧縮率の定義[4]の式を(1)に示す。

$$CR = \frac{Dr}{Dc} \cdots (1)$$

$Dc$  は符号化前のデータ

$Dr$  は符号化後のデータ

式(1)の圧縮率は、符号化後のデータのビット数と符号化前のデータのビット数の比で定義されている。CR の値が小さいほど高圧縮率である。CR が 0.01 なら元データから 1/100 に圧縮されていることを示している。

##### 5-2 $\gamma$ 符号化と $\delta$ 符号化の圧縮率の比較

図 8 に重み付き刺激データに対して  $\gamma$  符号化を用いた時の圧縮率を示す。図 9 に重み付き刺激データに対して  $\delta$  符号化を用いた時の圧縮率を示す。図 8、図 9 の横軸は刺激データの 14000 フレーム中のフレーム番号を示し、縦軸は刺激データの圧縮率を示す。加算するフレームの枚数が多いほど、圧縮率が高い。

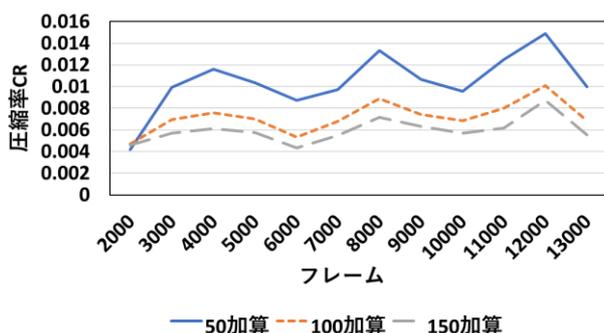


図 8 重み付き刺激データに対して  $\gamma$  符号化を用いた時の圧縮率

$\gamma$  符号化を用いた時は圧縮率が最も悪い場合 0.015 である。そして加算するフレーム数を増やすほど圧縮率が高まる。

$\gamma$  符号化の圧縮率 CR は平均で 0.00788 である。

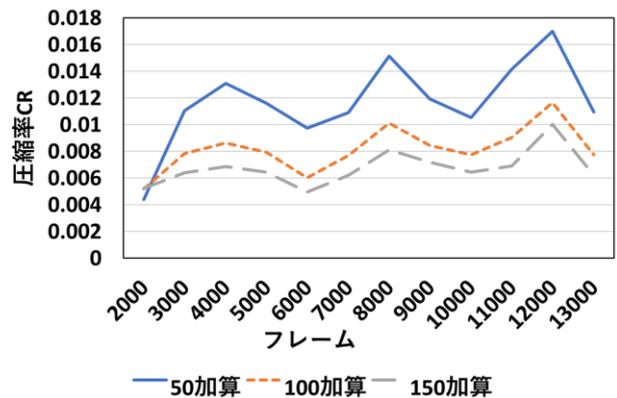


図 9 重み付き刺激データに対して  $\delta$  符号化を用いた時の圧縮率

$\delta$  符号化を用いた時は圧縮率が最も悪い場合 0.017 である。加算するフレーム数を増やすほど圧縮率が高まる。 $\delta$  符号化の圧縮率 CR の平均は 0.00889 である。

$\gamma$  符号化と  $\delta$  符号化を比較すると  $\gamma$  符号化の方が高圧縮であることがわかる。

#### 6. 考察

##### 6-1 $\gamma$ 符号化、 $\delta$ 符号化を用いた結果の比較

図 8 と図 9 より、重み付き刺激データに対して  $\gamma$  符号化の方が効率よく符号化できる。これは、図 5、図 6、図 7 より、重み付き刺激データの特徴として出現するデータの値が 1 や 2 など小さなものが大半であり大きくてもデータの値は 10 までしか出てこない特徴がある。そして、図 2、図 3 より整数値 10 以下の値には  $\gamma$  符号化の方が少ないビット数で表現することができることがわかっている。そのため、重み付き刺激データに対しては  $\delta$  符号化よりも  $\gamma$  符号化の方が少ないビット数で効率的に重み付き刺激データを表現することができる。

##### 6-2 フレーム加算枚数

本研究では、フレーム加算を行い、重み付き刺激データを作成してまとめて処理することによりデータの圧縮率を高めている。図 5-1 や図 5-2 から加算するフレームの枚数を増やすほど圧縮率が高まることがわかる。しかし、本論文の提案手法は可逆圧縮ではないのでフレームの加算枚数を増やすほどデータの損失率が大きくなる。このことから、データの損失率を調べるによりフレームの加算枚数を検討する必要がある。

#### 7. まとめ

視覚野刺激型人工視覚システムの刺激データをフレーム加算することにより重み付き刺激データを作成しても出現するデータの値は 1 や 2 などの小さな値が大半である。したがって、重み付き刺激データに対して  $\gamma$  符号化を用いた時の圧縮率 CR の平均は 0.00788 であり、 $\gamma$  符号化を用いた時の圧縮率 CR の平均は 0.00889 であり、 $\gamma$  符号化の方が高圧縮率を実現できた。今後の方針として、重み付き刺激データの特徴に効率的な符号化を検討する。本報告では重み付き刺激データを、そのままランレングス符号化を用いて得られた整数値のデータに対して  $\gamma$  符号化や  $\delta$  符号化を用

いている。しかし、重み付き刺激データの特徴として、隣接するデータ間には同じデータ値を有する性質がある。そのため、重み付き刺激データの隣同士の値で差分を取ることによりデータ量を減らし、圧縮率を高めることができる。と予想され、重み付き刺激データの差分データを作成することにより圧縮率を高める研究を進めていく。

## 謝辞

本研究は JSPS 科研費 JP20H00606 の助成を受けている。

## 参考文献

- [1] 八木哲也, 亀田成司, 飯塚邦彦, “可変受容野を備えた超並列アナログ知能視覚センサ”, 電子情報通信学会論文誌, D-I, 情報・システム, I-コンピュータ, Vol. 81, No. 2, pp. 104-113, 1998.
- [2] T. Sugiura, S. Nakatsuka, J. Yu, Y. Takeuchi, and M. Imai, “An efficient data compression method for artificial vision systems and its low energy implementation using ASIP technology”, Proceedings of IEEE Biomedical Circuits and Systems Conference, pp. 81-84, 2014.
- [3] Andrea Dierna, Meagan White, Danielle S. Miller, Eric DeCicco, “Introduction to DATA COMPRESSION”, Morgan Kaufmann, pp. 199, 2012.
- [4] P. Elias, “Universal codeword sets and representations of the integers”, IEEE Transactions on Information Theory, Vol. 21, No. 2, pp. 194-203, 1975.
- [5] A. Alarabeyyat, S. Al-Hashemi, T. Khmour, M. Hjouj Btoush, S. Bani-Ahmad, R. Al-Hashemi, “Lossless Image Compression Technique Using Combination Methods”, Journal of Software Engineering and Applications, Vol. 5, No. 10, pp. 752-763, 2012.
- [6] M. ネルソン, 「データ圧縮ハンドブックマルチメディアデータ圧縮の実践的プログラム技法」, ピアソンエデュケーション, 1996.