



Title	算術符号とその画像符号化への応用に関する研究
Author(s)	中井, 優一
Citation	大阪大学, 1990, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2432
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

算術符号とその画像符号化への 応用に関する研究

平成1年12月

中 井 優 一

この論文を両親に捧げる

内容梗概

本論文は、著者が大阪大学大学院工学研究科（通信工学専攻）在学中に行った研究の成果をまとめたものである。本論文は、以下の5章より構成されている。

第1章は序論であり、本研究に関連する分野において従来行われてきた研究について概括し、本研究の意義・所在を明らかにしている。

第2章では、本論文で扱っている算術符号について簡単に触れ、通信路において生じた誤りを検出する能力を有する無符号化区間挿入法と呼ぶ新たな算術符号の構成法を提案している。まず、無符号化区間挿入法の概念について述べ、次にそれを実現するための具体的な符号化・復号化アルゴリズムを示す。さらに、無符号化区間挿入法を用いた場合の平均符号長および誤り検出確率の評価式を導出している。また、計算機シミュレーションにより無符号化区間挿入法の誤り検出特性を明らかにしている。

第3章では、第2章で提案した無符号化区間挿入法を画像符号化に応用する場合について考察を行っている。まず、無符号化区間挿入法の誤り検出特性をより詳細に解析することによりこれを明らかにしている。次に、その結果に基づき無符号化区間挿入法の誤り検出能力を有効に発揮させるための新たな内挿法として部分内挿法と呼ぶ手法を提案する。さらに、実際の2値画像を用いて計算機シミュレーションを行い、部分内挿法を用いることにより画質の改善が図れることを示している。

第4章では、算術符号の濃淡画像符号化への一応用として分解能成分近似ブロック符号化法と呼ぶ新たな符号化法の提案を行っている。本符号化法は符号化すべき画像情報の局所的な性質に従って符号化モードを適宜切り替えて効率的な符号化を行う方式であり、符号化モード情報を符号化する際に算術符号

を用いることによって符号化特性の向上を図っている。また、実際の濃淡画像を用いたシミュレーションの結果、広い範囲の符号化レートにおいて従来の方式を上回る性能が得られることを述べている。

第5章は結論であり、本研究で得られた成果について総括を行っている。

目次

内容梗概	i
第 1 章 序論	1
第 2 章 誤り検出機能を付加した算術符号化法	7
2.1 緒言	7
2.2 算術符号	7
2.3 無符号化区間挿入法	9
2.3.1 無符号化区間挿入法の概念	9
2.3.2 符号化・復号化アルゴリズム	11
2.3.3 平均符号長および誤り検出確率	15
2.4 計算機シミュレーション	16
2.5 結言	22
第 3 章 無符号化区間挿入法の画像符号化への応用	25
3.1 緒言	25
3.2 部分内挿法	25
3.3 計算機シミュレーション	29
3.4 結言	36
第 4 章 分解能成分近似ブロック符号化法	37

4.1	緒言	37
4.2	ブロック符号化法	37
4.3	分解能成分パターンを用いたブロック符号化	41
4.3.1	分解能成分パターンの選択	41
4.3.2	選択された分解能成分パターンを用いた符号化	45
4.3.3	符号化モード	48
4.3.4	階調成分の符号化	49
4.3.5	算術符号によるモード識別フラグの符号化	50
4.4	計算機シミュレーション	52
4.5	結言	54
第 5 章 結論		55
謝辞		57
参考文献		58
付録 無符号化区間挿入法を用いた場合の平均符号長の導出		67

第 1 章

序論

古来より、画像を用いたコミュニケーションは人間の視覚に直接的に訴えてくる形態であり、またその内容を理解するのに特別な訓練を必要としない、いわゆる「わかりやすい」情報伝達手段であるために様々な場所で盛んに用いられてきている。しかしながら、画像を用いた通信と言うことになると広く普及しているのはせいぜいテレビジョン放送あるいはファクシミリ通信程度であろう。優れた情報伝達手段でありながら画像通信が広く普及しない理由としては、一般に画像の持つ情報量が膨大なものであるため原情報の形のままで通信路を伝送するには広大な周波数帯域幅を要し、またコストあるいは処理時間がかかり過ぎるということがあげられる。従って、種々の画像通信が実用化されるか否かは伝送すべき情報量をどこまで削減できるかにかかっていると言うことができよう。

この問題に対して 1948 年に Shannon は彼の著名な論文^[1]において、情報源より生起するシンボルを符号化する際の平均符号長の下限は情報源のエントロピーに等しいことを証明し、情報源符号化に対する理論的なよりどころを示した。さらに、1952 年 Huffman^[2] は情報源シンボルの生起確率が既知である情報源に対する具体的な最適符号構成法を提案している。この 2 つの先駆的な

研究によって我々は、情報源モデルが与えられれば達成し得る最短の平均符号長を知ることができ、必要であれば平均符号長が情報源のエントロピーと同程度となる符号を構成することが可能となった。また、同年 Kretzmer^[3] がテレビジョン信号には多くの冗長性が存在することを指摘し、画像符号化による情報圧縮の可能性を示唆している。これらの技術により少なくとも理論的には画像符号化の問題は解決されたかに見える。しかしながら、これらは必ずしも実際問題としての画像符号化に対する解決法を与えるものとは成り得なかった。なぜなら、これらの議論は情報源モデルが与えられているという前提の上に成り立つものであるが実際には種々の画像を効率よくモデル化できるような情報源モデルを見いだすこと自体が一般には非常に困難だからである。ファクシミリ通信等で対象とする線画像等に対しては周辺画素によって条件付けられる m 重マルコフモデルが比較的によく用いられているが、本来非定常な情報源である画像を表すのには不十分であり、濃淡画像に対するモデルとしてはより一層不適当である。この問題に対して近年、隠れマルコフモデル^{[4]–[7]} と呼ばれるモデルを用いて画像の情報源モデルを構築しようとする研究^{[8]–[10]} も行われているが十分な成果をあげるには至っていない。そのような事情から画像符号化の研究は、主に画像の定性的な性質を捉えそれに基づいて符号化を行う方式の研究が主流となっている。しかしながら、現在までになされたいずれの研究も、得られる性能あるいは処理量・装置規模等の点で一長一短でありほとんどのものは実用化に至っていない。

しかし、近年の半導体技術の急速な発展によって小型で大容量のメモリーが容易に利用できるようになり、もはや「大容量のメモリーを必要とする」ということは実用化への障害とはなり得なくなると言ってもよいであろう。また、小型でより高速な処理装置が得られるようになったため、従来は不可能であると思われていた複雑で高度な符号化処理も行えるようになる等、画像符号

化をとりまくハードウェア環境は大きく変化しようとしている。また、応用面では現在構築されつつある総合サービスディジタル通信網 (ISDN) を用いたテレビ会議・テレビ電話、ビデオテックスに代表される画像情報検索サービスあるいは次世代のテレビジョンとして期待される高精細度テレビジョン (HDTV) 等の有望なメディアの実用化への動きがあり、画像通信に対する需要は飛躍的に高まる事が予想される。それに伴い、画像符号化あるいはそれを包含する情報源符号化の果たす役割は今後ますます重要なものとなることが予想される。また、CD-ROM 等の新しい媒体を用いた大容量の画像蓄積・検索システムの登場により通信以外の新たな用途においても画像符号化の必要性が高まってきている。

一般に情報源符号化は、復号された情報が送信された情報と必ずしも完全に一致しないような方式 (歪のある符号化方式) と、復号した時に送信された情報と全く同じ情報が得られるような方式 (歪のない符号化方式) に大別することができる。前者に属する符号化方式としてはベクトル量子化^{[11]–[14]}、変換符号化^{[15]–[16]}あるいは本論文第4章でも取り上げるブロック符号化^{[50]–[59]}等がある。また、後者に属する情報源符号化としては従来より Huffman 符号化がよく知られているが、近年算術符号化と呼ばれる符号化方式が注目を集め、盛んに研究がなされている^{[17]–[47]}。

算術符号化は特定の符号化アルゴリズムを指すものではなく P.Elias によって提案された符号化原理^[17]に基づく符号化法の総称である。Elias の符号化原理は非常に単純ではあったが符号化時に無限精度の演算を必要とするという欠点を有しており、そのままでは実用化できないものであった。しかし、その後 1976 年に Pasco^[18] が Elias の方法を比較的忠実な形で実現するアルゴリズム

ムを提案した。また、同年 Rissanen^[19] は従来から知られている Kraft の不等式の拡張という立場から算術符号化のアルゴリズムを提案した。この年を契機として算術符号の優れた性質が注目され始め、活発に研究が行われるようになった。以後 Rubin^[20] , Guazzo^[21] あるいは Jones^[22] 等によって様々な形の算術符号が提案されている。算術符号化は、単純な算術演算で符号化・復号化が実行できる、Huffman 符号化等とは異なり情報源シンボル当たりの符号化レートを情報源のエントロピーに近づけるために情報源の拡大を必要としないこと、さらに情報源シンボルの生起確率の変化に対する追従性に優れているということ等の特長を有する優れた符号化法である。このため、2 値画像符号化^{[24]–[25]} あるいは濃淡画像符号化^{[26]–[27]} への応用、さらに音声符号化^[28] への応用も行われている。また、適切な情報源モデルを想定できないような情報源からの出力を符号化する場合に、符号化の過程で情報源の確率パラメータを逐次的に学習しながら符号化を行う、いわゆるユニバーサル符号化の手法としても用いられている^{[29]–[31]}。本論文は、このような特長を持つ算術符号の画像符号化への応用という観点から様々な考察を行ったものである。

算術符号は、処理量あるいはその性能面において極めて有望な情報源符号化の手法であるが実用化を阻む欠点として誤り伝播の問題を有している。算術符号は、非ブロック符号であるため通信路において符号語に誤りが生起すると復号された情報において誤りが無限に伝播して行くという欠点を有している。一方、Huffman 符号はブロック符号であるため誤り伝播に関しては算術符号に比べて対処しやすく、誤りが生起しても誤りが無限に伝播しないような符号構成法が古くから提案されている^{[48]–[49]}。それに比較して算術符号の誤り伝播の問題は、算術符号が非ブロック符号であるため取り扱いがむずかしく、現在までほとんど研究がなされていない^[40]。そこで第2章では、この誤り伝播の問題に一つの解決法を与える無符号化区間挿入法と呼ぶ新たな算術符号の構成

法を提案している。無符号化区間挿入法は従来の算術符号の符号化・復号化アルゴリズムを一般化することによって実現しており、算術符号の特長を損なうことなく誤り検出機能を付加した方式である。まず、無符号化区間挿入法の概念について述べ、次に具体的な符号化・復号化アルゴリズムを示す。また、平均符号長および誤り検出確率の評価式の導出を行っている。さらに、計算機シミュレーションによって無符号化区間挿入法の性能評価を行う。

第3章では、第2章で提案した無符号化区間挿入法を画像符号化へ応用する場合に有効となる新たな内挿法を提案している。一般に画像符号化において誤り検出能力を利用して画質の劣化を抑制する手法としては誤りを検出した走査線を直前の走査線で置き換える走査線内挿法がしばしば用いられるが、無符号化区間挿入法においては誤りを検出した位置の近傍までは正しく復号を行っているという性質を有しているため、それを利用した部分内挿法なる内挿法を提案する。これは、走査線内挿を行う場合、直前の走査線すべてを内挿するのではなく誤りを検出した位置の近傍までは復号結果をそのまま用い、それ以後を直前の走査線から内挿するものである。この手法を用いることによってより簡単な構成で従来方式と同等の画質が得られることを実際の2値画像を用いた計算機シミュレーションによって確認している。

また、算術符号を濃淡画像の符号化に応用した研究^{[26]–[27]}はあるが、一般に歪のない符号化方式に比較して歪のある符号化方式の方が高い圧縮率を得られるために、実用上特殊な用途を除いては濃淡画像の符号化には歪のある符号化方式が用いられる場合がほとんどである。そのような符号化においては従来より部分的に Huffman 符号が用いられることはあったが算術符号の応用例はほとんど存在しない。しかしながら、算術符号は情報源の確率パラメータを前もって知らなくても符号化できるという利点を有していることから、明確な情報源モデルが存在しない画像の符号化への応用は Huffman 符号等より適してい

と考えられる。そこで第4章では算術符号の濃淡画像符号化への一応用として分解能成分近似ブロック符号化法と呼ぶ符号化法の提案を行っている。分解能成分近似ブロック符号化法は画像の局所的な性質に応じて符号化モードを適宜切り替えながら符号化を行う方式であるが、この符号化モード情報の符号化に算術符号を適用することによって符号化特性の向上を図ったものである。濃淡画像を用いた計算機シミュレーションにより、広い範囲の符号化レートにおいて従来方式を上回る性能が得られることを述べている。

最後に、第5章では結論として本研究で得られた成果について総括している。

第 2 章

誤り検出機能を付加した算術符号化法

2.1 緒言

本章においては，本研究で取り扱う算術符号について述べ，通信路において生起した誤りを検出する能力を有する無符号化区間挿入法^{[41]–[47]}と呼ぶ新たな算術符号を提案する．また，無符号化区間挿入法を用いた場合の平均符号長および誤り検出確率の評価式を導出する．さらに，計算機シミュレーションによって無符号化区間挿入法の誤り検出特性の評価を行う．

2.2 算術符号

P.Elias によって提案された算術符号の符号化原理^[17]は，数直線 $[0,1)$ 区間を入力シンボルの生起確率の比に従って逐次分割していき最終的に一つの入力シンボル列に対して $[0,1)$ 区間内のある部分区間を割り当て，符号語としてはその部分区間内に存在する任意の点の数値を 2 進数展開して用いるという非常に単純な方法である．たとえば図 2.1 の例では，情報源が無記憶 2 値情報源であり，“0”の生起確率を $P(0)=0.7$ ，“1”の生起確率を $P(1)=0.3$ とした場合の入力シンボル列 $s=“010”$ に対する $[0,1)$ 区間の分割の様子が示されている．まず $[0,1)$ 区間をシンボルの生起確率の比 7:3 に分割する．最初の入力シンボルが

“0”であるので、区間 $[0, 0.7)$ が選択される。さらに、 $[0, 0.7)$ を 7:3 に分割し次の入力シンボルが “1” であるので区間 $[0.49, 0.7)$ が選択される。同様に $[0.49, 0.7)$ を 7:3 に分割し、最後の入力シンボルが “0” であるので区間 $[0.49, 0.637)$ が選択される。符号語は $[0.49, 0.637)$ 内の任意の数値を 2 進数展開して得られる。従って、算術符号の符号語は $[0, 1)$ 区間内のある点の数値を表していると言える。復号は符号語の表わす数値が何れの部分区間に属するかを判定することによって行われる。

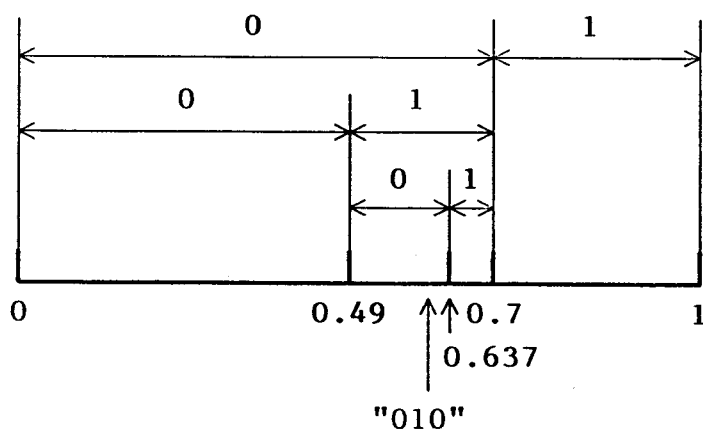


図 2.1: 算術符号の原理

この方法は符号化が単純な算術演算で実行可能な方法であるが、入力シンボル列が長くなるにつれて必要とされる演算精度が無限に増加して行くという欠点を有していた。しかし、その後 1976 年に Pasco^[18] が Elias の方法を比較的忠実な形で、しかも有限長の演算精度で実行可能な実用的なアルゴリズムを提案した。また、同年 Rissanen^[19] は従来から知られている Kraft の不等式の

拡張という立場から算術符号化のアルゴリズムを提案した。2.3.1で述べる様に、本章において提案する手法は、従来より提案されている算術符号の符号化・復号化アルゴリズムに修正を加え、かつ一般化することによって実現することができる。基本となる符号化・復号化アルゴリズムの選択については必ずしも制約は存在しないが、本論文では、Elias の符号化原理を最も忠実に実現している Pasco の算術符号化のアルゴリズムに基づいて新しい手法を提案する。

2.3 無符号化区間挿入法

2.3.1 無符号化区間挿入法の内容

2.2で述べたように算術符号の符号語は $[0,1)$ 区間内のある数値を表していると考えることができる。従って、この符号語に誤りが生起するということは符号語の表している数値が変化することと等価である。

従来の算術符号化では、 $[0,1)$ 区間を隙間なく分割しそれぞれの部分区間を異なった入力シンボル列に割り当てて符号化を行っている (図 2.2) ため、 $[0,1)$ 区間内のあらゆる点はいずれかの入力シンボル列に割り当てられている部分区間内に存在している。従って、符号語に誤りが生起してその数値が変化しても必ずこれに対応する別な入力シンボル列が存在するために誤った入力シンボル列が復号される。

そこで、 $[0,1)$ 区間を分割する際にいずれの入力シンボル列にも属さない部分区間 (以後、この部分区間を無符号化区間と呼ぶ。) を設けることを考えてみる (図 2.3)。このような無符号化区間を設けた場合には、誤りによって符号語の表す数値が変化しても別の入力シンボル列に割り当てられている部分区間内に遷移するとは限らない。従って、誤りによって符号語の表す数値が無符号化区間内に遷移した場合には復号すべき入力シンボル列が存在しないことから誤りが生起していると判定することができる。たとえば、入力シンボル列 A , B

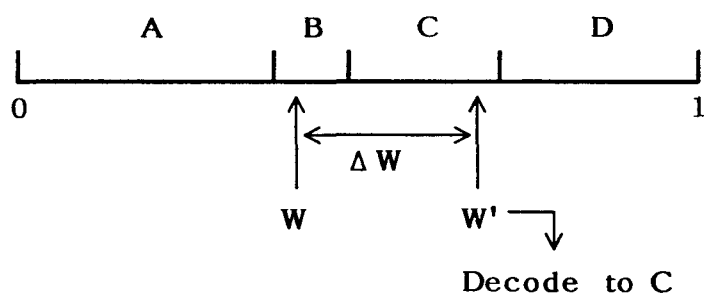


図 2.2: 通常の算術符号

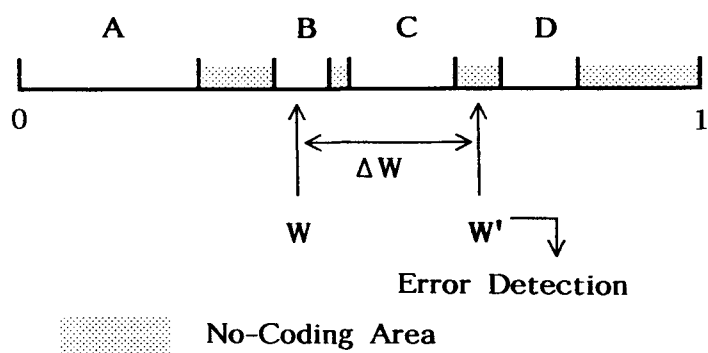


図 2.3: 無符号化区間挿入法

， C ， D に対して図 2.2， 図 2.3 のように部分区間を割り当てたと仮定する．
 この時， 符号語 W に誤りが生起して符号語の大きさが ΔW だけ変化し W' になった場合を考える． 図 2.2 においては W' はシンボル列 C に割り当てられた部分区間内に存在するために C へ誤って復号される． しかしながら， 図 2.3 の場合には符号語 W' が無符号化区間内に存在することから誤りが生起していると判定することができる． 以上に述べた手法を以後， 無符号化区間挿入法と呼ぶことにする．

2.3.2 符号化・復号化アルゴリズム

本節においては， 無符号化区間挿入法を実現するための符号化・復号化アルゴリズムを示す． なお， 簡単のため以下においては， 情報源は一重マルコフ 2 値情報源を仮定する．

今， 長さ N の入力シンボル列

$$X(N) = x_1 x_2 \cdots x_N \quad x_i \in \{0, 1\}$$

を符号化することを考える． 符号化に用いるレジスタは長さ L_F の F レジスタおよび長さ L_T の T レジスタの 2 つのレジスタであり， いずれも最上位ビットが小数第 1 位であると仮定する． また， $P(i|j)$ はシンボル j が生起した条件のもとでのシンボル i が生起する条件付き確率， $C(i|j)$ はシンボル i の条件付き累積確率を表し，

$$C(i|j) = \sum_{n < i} P(n|j) \quad (2.1)$$

で定義される． $P(i|j)$ ， $C(i|j)$ は F あるいは T レジスタと同様に最上位ビットが小数第 1 位で各々長さ L_P で表されているものとする． $C(\cdot)$ ， $P(\cdot)$ はそれぞれ， 各シンボルに割り当てられる部分区間の始点および区間幅に対応する． また， $\langle \cdot \rangle_m$ は有効桁数 m 桁での打ち切りを表すものとする．

2.3.1で述べたように，誤りを検出するためには何らかの方法を用いて無符号化区間を設定すればよい．無符号化区間を設定する方法は任意であるが，本研究では従来からの算術符号の符号化・復号化アルゴリズムを一般化するという立場から，Pascoによって提案されたアルゴリズムに対して，シンボル j が生じた条件のもとでのシンボル i の部分区間の短縮率を導入することによって無符号化区間の設定を行っている．この短縮率を $k(i|j)$ と記すが，この $k(\cdot)$ によって無符号化区間の制御を行うことができる．なお，この $k(\cdot)$ は

$$0 < k(\cdot) \leq 1 \quad (2.2)$$

を満たす実数である．のちほど明らかにするように短縮率 $k(\cdot)$ は無符号化区間挿入法の性能を決定する重要なパラメータである．この $k(\cdot)$ を 1 とすることにより，以下に述べるアルゴリズムは従来のもものと一致する．このような意味から，以下の符号化・復号化アルゴリズムは従来アルゴリズムを一般化したものとみなすことができる．

《符号化アルゴリズム》

1. $F \leftarrow 0$, $T \leftarrow 1$ とする．
2. 入力シンボル x_i ($i = 1, 2, \dots, N$) に対して 2a ～ 2b を行う．

$$2a. F \leftarrow F + T \cdot C(x_i|x_{i-1})$$

$$2b. T \leftarrow \langle T \cdot P(x_i|x_{i-1}) \cdot k(x_i|x_{i-1}) \rangle_m$$

この時 T の最上位ビットが 0 になれば，最上位ビットが 1 になるまで T をシフトし，同時に F も同一ビット数だけシフトする．そして， F から桁あふれする内容を符号語 W として通信路に送り出す．

3. F を1ビットシフトして1を加える.

次に復号化アルゴリズムについて述べる. 復号化においては F および T レジスタの他に新たに F , T と同様の構造を持つ長さ L_G のレジスタ G を必要とする. また, 符号語 W の先頭から n ビットの部分列を $W(n)$ で表すものとする.

《復号化アルゴリズム》

1. $F \leftarrow W(L_F)$, $T \leftarrow 1$ とする.
2. $i = 1, 2, \dots, N$ に対して 2a ~ 2d を行う.

$$2a. G \leftarrow F/T$$

2b.

$$C(y_i|y_{i-1}) \leq G < C(y_i|y_{i-1}) + P(y_i|y_{i-1}) \cdot k(y_i|y_{i-1})$$

を満たす $y_i \in \{0, 1\}$ を復号シンボルとして出力する. そのようなシンボルが存在しない場合には, 誤りが生起したと判定し復号を終了する.

$$2c. F \leftarrow F - T \cdot C(y_i|y_{i-1})$$

$$2d. T \leftarrow \langle T \cdot P(y_i|y_{i-1}) \cdot k(y_i|y_{i-1}) \rangle_m$$

この時 T の最上位ビットが0になれば, 最上位ビットが1になるまで T をシフトし, 同時に F も同一ビット数だけシフトし, そのビット数だけ符号語を読みこむ.

ただし x_0 , y_0 は共に最も生起確率の高いシンボルを表すものとする。

ここで,

$$P'(i|j) = P(i|j) \cdot k(i|j) \quad (2.3)$$

とおくと $k(\cdot)$ は式 (2.2) から分かるように 1 以下の数値であるので,

$$P'(i|j) \leq P(i|j) \quad (2.4)$$

である。 $P'(i|j)$ は物理的には部分区間の幅に対応するのでシンボル i には本来割り当てられるべき部分区間よりも短い部分区間しか割り当てられない。その際、割り当てられなかった部分区間が無符号化区間となるわけである。また、算術符号の原理から、以上のような操作を行っても通信路で誤りが生起しない場合には正しい復号が行われることは明らかである。

次に、符号語に誤りが生起した場合について復号化アルゴリズムに従って考察する。今、受信語の第 i ビット目に誤りが生起していると仮定する。復号化アルゴリズムから明らかなように、第 i ビット目が F レジスタに読み込まれるまでは F および T レジスタの内容の遷移は符号化時のものと同一であり正しい復号が行われる。しかしながら、 F レジスタに第 i ビット目が読み込まれた後の F , T レジスタの内容の遷移の様子は符号化時のものと異なっている。 $k(\cdot) = 1$ の場合は F , T レジスタの遷移が正しく行われていなくてもそのことを検知できずに誤った復号を続けて行くことになる。すなわち、受信語に誤りが生起した場合にはあるビットを先頭として誤りが伝播していくこととなる。しかしながら一般に $k(\cdot) < 1$ の場合には、復号を行っていくに従い復号化アルゴリズム 2b. の式を満たさない場合が生じ、従ってこの場合には誤りを検出す

ることができる。以上より、本アルゴリズムを用いた場合には誤りを検出する位置 S_d と復号結果が元の系列と比較して初めて差を生ずる位置 S_c の間には、誤りを検出した時点で復号を終了することを考慮すると、

$$\frac{S_c}{S_d} \leq 1 \quad (2.5)$$

なる関係が成立することが明らかである。

本手法においては、誤り検出確率および符号長は短縮率 $k(\cdot)$ にのみ依存しておりこの $k(\cdot)$ を変化させることによって連続的に誤り検出確率・符号長を変化させることができる。しかも、これは通信路符号化とは全く独立の処理であるから通信路符号化と組み合わせて使用することは可能であり、その場合にはより自由度に富んだ冗長度・誤り検出確率の設定を行うことができる。

2.3.3 平均符号長および誤り検出確率

本節では、無符号化区間挿入法を用いた場合の平均符号長、誤り検出確率の評価式を導出する¹。平均符号長の評価式 L_{av} は Pasco と同様の方法によって導くことができ、次のように表わせる。

$$NH_P + 1 + \Delta \leq L_{av} < NH_P + NV_2(m) + 2 + \Delta \quad (2.6)$$

ただし、

$$\Delta = - \sum_{X(N)} P(X(N)) \sum_{i=1}^N \log k(x_i | x_{i-1}) \quad (2.7)$$

$$V_2(m) = -\log(1 - 2^{1-m}) \quad (2.8)$$

であり、 H_P は情報源のエントロピー、 $P(X(N))$ は系列 $X(N)$ の生起確率を表す。 Δ は誤り検出能力を得るために付加した冗長分と見なすことができる。

¹詳細は付録 A を参照のこと

次に、誤り検出確率の評価式を導出する。生起する誤りによって符号語の表す大きさがランダムに変化すると仮定すると、誤り検出確率は無符号化区間が $[0,1)$ 区間内に占める割合として与えられる。従って一重マルコフ情報源の場合には誤り検出確率 P_D は次式のように与えられる。

$$P_D = 1 - \sum_{X(N)} \prod_{i=1}^N k(x_i|x_{i-1})P(x_i|x_{i-1}) \quad (2.9)$$

2.4 計算機シミュレーション

本節では、計算機シミュレーションにより無符号化区間挿入法の誤り検出特性を評価する。まず、無符号化区間挿入法において符号語に単一誤りが生起した場合の誤り検出確率と符号長 L の関係についてシミュレーション実験を行う。

その結果を図 2.4 に示す。図において L_0 は $k(\cdot) = 1$ の時の符号長を表す。この図より、ほんのわずかの冗長で良好な誤り検出確率が得られていること、また整数値ではなく連続的に冗長が与えられることがわかる。

次に、本手法を他の符号化手法と比較しつつ評価する。なお、比較のための符号化法としては代数的手法 (ガロア体 $GF(2)$ 上で多項式表現した算術符号の符号語を、 $GF(2)$ 上の既約多項式 $G(X)$ で割った余りを検査記号として付加する方法。巡回符号ではないことに注意されたい。) を考える。そして以下の手法、すなわち

1. 無符号化区間挿入法
2. 代数的手法
3. 無符号化区間挿入法と代数的手法を組み合わせる方法 (ハイブリッド法と呼ぶ)

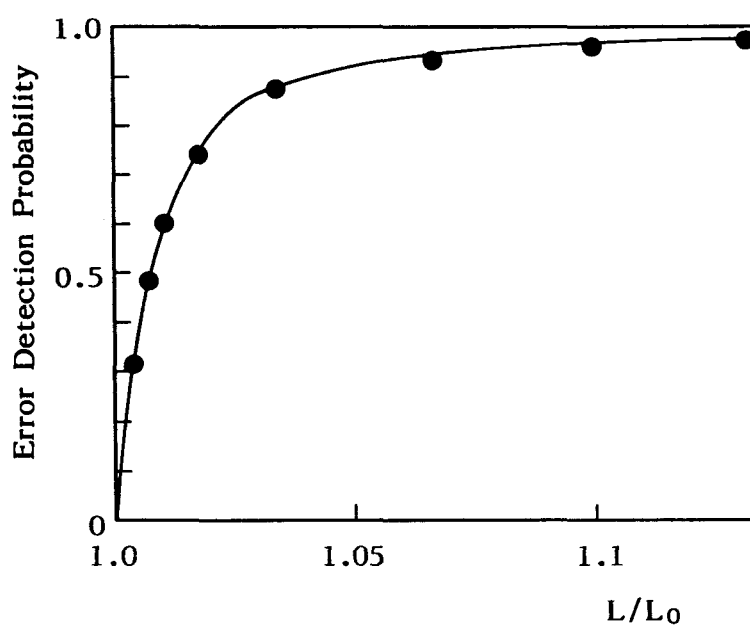


図 2.4: 誤り検出確率 (単一誤り)

について、冗長ビット数 δ を 0.5 から 3 ビットまで変化させた場合の誤り検出確率 P_D を通信路誤り率 ε の関数として求める。ただし、冗長ビット数 δ は代数的手法においては 1 ビット毎、無符号化区間挿入法およびハイブリッド法については 0.5 ビット毎に変化させる。また、通信路としては誤り率 ε の 2 元対称通信路を仮定し、符号化・復号化で使用する各パラメータの値は表 2.1 のように定める。また、シミュレーションに用いた情報源出力系列の数は 600 である。

表 2.1: シミュレーションに用いたパラメータの値

パラメータ	値
$P(0 0)$	0.9
$P(1 0)$	0.1
$P(0 1)$	0.2
$P(1 1)$	0.8
N	540
L_F	62
L_T	31
L_G	31
L_P	31
m	31

冗長ビット数 δ および通信路誤り率 ε を種々の値にした場合の結果を各々図 2.5～図 2.7 に示す。

これらの結果より次のことが明らかである。

1. 図 2.7 の $\delta = 2, 3$ (ビット) の場合から分かるように冗長ビット数が同一の場合は、代数的手法とハイブリッド法はほぼ同一の誤り検出確率を与える。

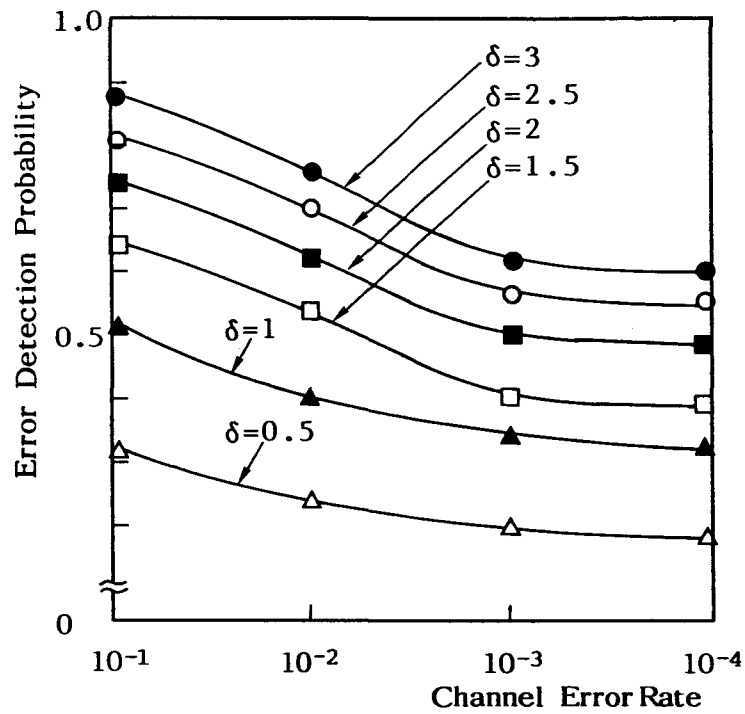


図 2.5: 誤り検出確率 (無符号化区間挿入法)

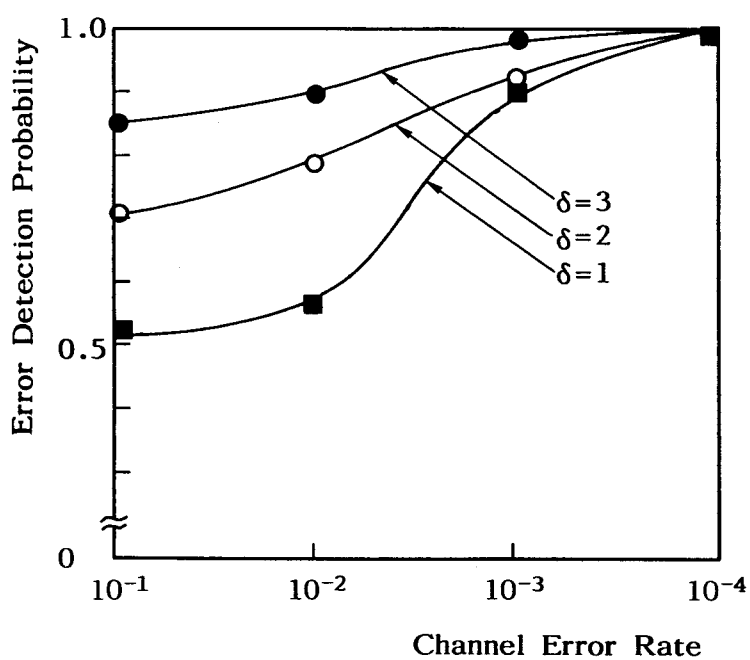


図 2.6: 誤り検出確率 (代数的手法)

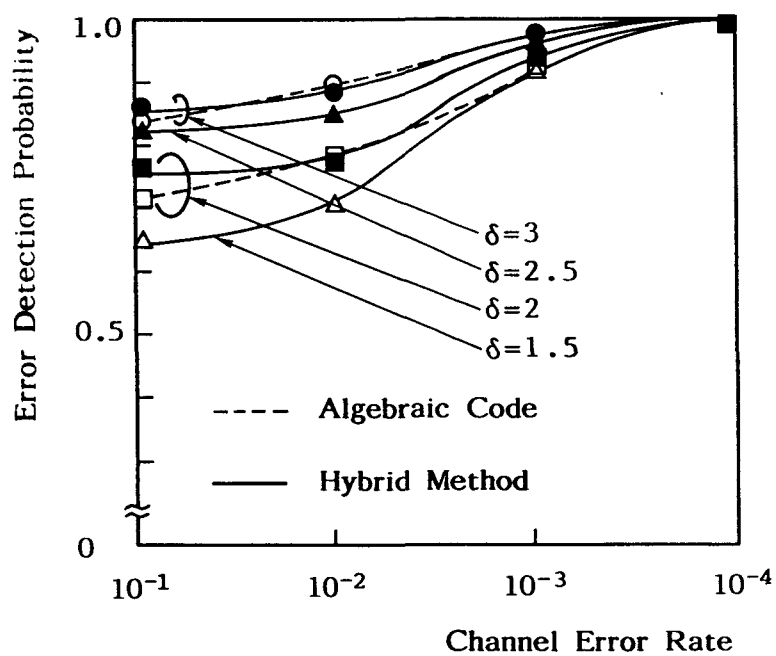


図 2.7: 誤り検出確率 (ハイブリッド法)

2. ハイブリッド法において冗長ビット数が整数でない場合の誤り検出確率は整数の冗長ビットを付加した代数的手法の中間的な誤り検出確率を与える。
3. 通信路誤り率が小さくなるにつれて代数的手法およびハイブリッド法では誤り検出確率が増加して行く。それに対して、無符号化区間挿入法の誤り検出確率は減少して行く。

1., 2. から明らかなようにハイブリッド法を用いることによって整数の冗長ビット数を持たせた場合には代数的手法と同じ性能を得ることができ、更に代数的手法単独では実現不可能であった連続的な冗長ビット数を与えることが可能であることが分かる、またその誤り検出確率が冗長ビット数に比例して増加して行くために容易に所望の誤り検出確率を得ることが可能であることが分かる。

3. については、明らかに無符号化区間挿入法にとって不利な性質であるが、この原因を詳細に追求してみると本手法を画像符号化に応用した場合、画面の歪をより改善する手法が存在することが明らかとなる。次章においてこのことを詳細に述べる。

2.5 結言

本章においては、通信路において生起した誤りを検出する能力を有する無符号化区間挿入法と呼ぶ新たな算術符号の構成法を提案し、具体的な符号化・復号化アルゴリズムを示すと共に、本構成法を用いた場合の平均符号長および誤り検出確率の評価式を導出した。

無符号化区間挿入法は従来の算術符号を一般化することによって実現でき、その特長を損なうことなく算術符号に誤り検出能力を付与したものである。さらに通常の通信路符号化と組み合わせることによってより自由度に富んだ冗長・

誤り検出確率の設定が可能であることを計算機シミュレーションによって明らかにした。

第 3 章

無符号化区間挿入法の画像符号化への応用

3.1 緒言

本章では，第 2 章で提案した無符号化区間挿入法を画像符号化へ応用する場合について考察を行っている．まず無符号化区間挿入法の誤り検出特性をより詳細に調査し，これを明らかにする．その結果に基づいて無符号化区間挿入法の誤り検出能力を有効に発揮させることのできる新たな内挿法として部分内挿法を提案する．また，実際の 2 値画像を用いて計算機シミュレーションを行いその有効性を明らかにする．

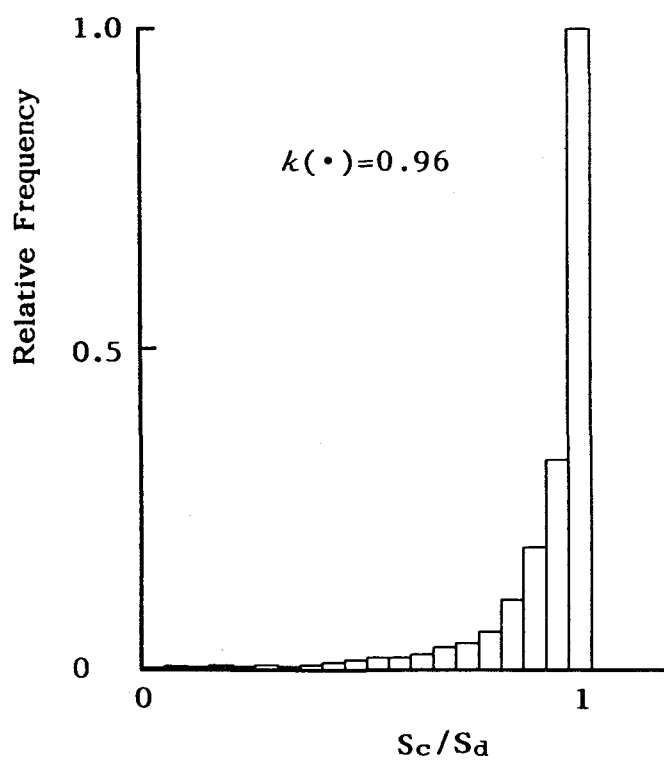
3.2 部分内挿法

第 2 章で指摘したように無符号化区間挿入法においては通信路誤り率が低くなるに従って誤り検出確率が減少して行くが，これは次のように説明することができる．すなわち，通信路誤り率が低くなるに従って通信路で生起する誤りの数は減少して行き，単一誤りが支配的となる．しかるに無符号化区間挿入法を用いる場合には，復号アルゴリズムから明らかなように誤り検出確率は誤りが生起したビットを F レジスタに読み込んだ後の復号回数にほぼ比例すると考えてよい．従って，通信路で生起した誤りが符号語の先頭に近いほど，また

情報源からの入力シンボル系列長が長くなるほど誤り検出が容易になるが、逆に誤りの生じる位置が後方になるに従い、また入力シンボル長が短くなるほど誤り検出は逆に困難になってしまう。つまり、無符号化区間挿入法においてはその誤り検出確率は誤りの生起した位置とそれ以後復号されるべきシンボル数の関数になると考えられる。従って、入力シンボル系列長が固定の場合には誤り検出確率は通信路において生起する誤りの位置のみに依存することとなる。単一誤りが支配的となる領域においてはこの傾向は一層顕著なものとなり無符号化区間挿入法の誤り検出確率は減少して行くことになる。しかしながら、以下に述べるように無符号化区間挿入法においては誤りを検出した場合にはそれ以前の復号結果の大部分は正しく復号されているために、本手法を画像符号化等に応用する場合には新たな内挿法を用いることによって、誤り検出確率が低下する領域においても誤り発生による平均的な歪を低減させることができ代数的手法を上まわるパフォーマンスが期待できる。

第2章において指摘したように、本手法においては誤りを検出する位置 S_d と復号結果が元の系列と比較して初めて差を生ずる位置 S_c の間には式 (2.5) の関係が成立する。そこで、 S_c と S_d の関係をより詳細に調査した。短縮率 $k(\cdot)$ に対する S_c/S_d の分布の例を図 3.1, 3.2 に示す。この結果より、 $k(\cdot)$ の値が大きくなるにしたがって拡がる傾向は示すもののその分布は $S_c/S_d = 1$ の近傍に集中していることが分かる。すなわち、無符号化区間挿入法においては誤りを検出した位置の近傍まで正しく復号を行っていると言うことができる。

一般に画像符号化において誤り検出能力を利用して画質の改善を行う場合には、符号化・復号化を走査線単位で行い、復号時に誤りを検出した走査線を直前の走査線で置き換えることによって画面の歪の増大を抑える走査線内挿法と呼ばれる手法がしばしば用いられている。これは、通常の誤り検出符号では誤りが生起していることは検出できても誤りの生起した位置までは検出できな

図 3.1: S_c/S_d の分布 ($k(\cdot)=0.96$)

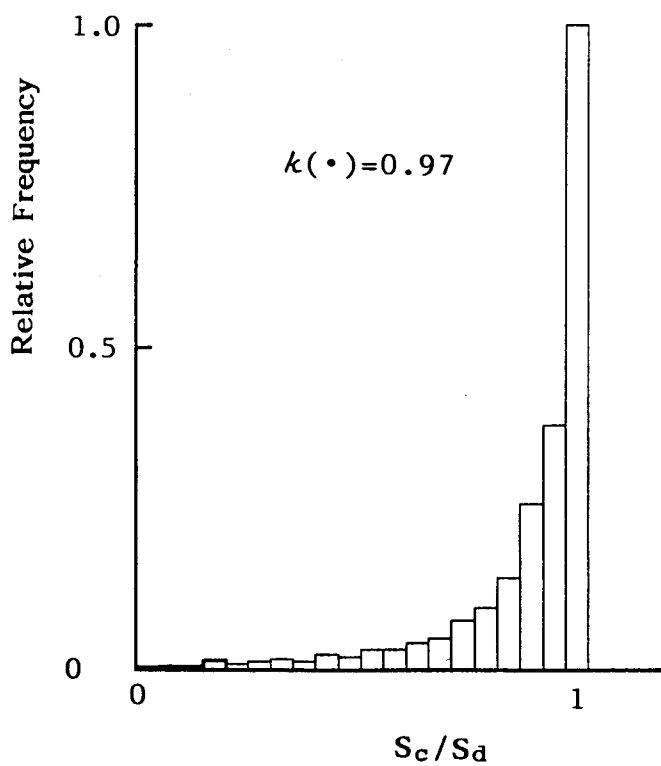


図 3.2: S_c/S_d の分布 ($k(\cdot)=0.97$)

いためである。誤りの生起した位置を知るためには、一つの走査線をさらに複数のブロックに分けて、その各々のブロックに誤り検出能力を与えるか、誤り訂正符号を適用する等の手段を用いなければならないが、いずれの場合でもより多くの冗長を付加する必要がある。しかるに、本手法では上述のように誤りを検出した位置の近傍までは正しい復号を行なっているという特長を有しているため、新たに冗長を付加しなくても誤りの生起した位置をほぼ推定することができる。従って、誤りを検出した場合には直前の走査線全部を内挿するのではなく、誤りを検出した位置の近傍まで復号結果をそのまま用いそれ以後を直前の走査線から内挿するという手法を用いることによってさらに画質の改善を図ることが可能となる。すなわち、代数的手法における誤り検出確率を P_{Da} 、誤りを見逃したことによって生じる歪を D_{aud} 、誤りを検出しかつ内挿を行ったことによる歪を D_{aip} とし、さらに部分内挿法を用いた場合の無符号化区間挿入法の誤り検出確率を P_D 、誤りを見逃したことによって生じる歪を D_{rud} 、部分内挿を行ったことによって生じる歪を D_{rip} とすると、 S_c/S_d の分布等を考慮すると、通信路誤り率の値によっては

$$P_{Da}D_{aip} + (1 - P_{Da})D_{aud} \geq P_D D_{rip} + (1 - P_D)D_{rud} \quad (3.1)$$

が成立することが予想される。また、この場合情報源符号化の過程で同時に誤り検出能力も付加しているために代数的手法のように改めて誤り検出能力を付与するための処理の必要がないことから全体の構成の簡単化あるいは処理時間の短縮を図ることができる。この手法を以後、部分内挿法と呼ぶ。

3.3 計算機シミュレーション

部分内挿法を画像符号化へ応用する場合には、再生される画像の歪は無符号化区間挿入法の誤り検出確率および誤りを検出した位置に対して復号結果をどこまで用いるかによって決定される。実際の2値画像「天気図」(540 × 600

画素) に対して部分内挿法を適用した結果を図 3.3 に示す. ただし図における歪 D および部分内挿率 I_p は次式で定義する.

$$D = \frac{N_d}{N_p} \quad (3.2)$$

$$I_p = \frac{S_i}{S_d} \quad (3.3)$$

ここで N_p は原画像の画素数, N_d は再生画像において原画像と異なる画素数, S_d は誤りを検出した位置を, S_i は部分内挿のために用いられる復号結果の最終の位置を表すものとする. 従って, 通常の誤り検出符号を用いた場合の走査線内挿法は部分内挿法において $I_p = 0$ とした場合と等価である. このような意味で, 部分内挿法は通常の走査線内挿法をその特殊な場合として含む, より一般的な内挿法であると言える.

図 3.3 より部分内挿率 I_p を適当に選ぶことによって部分内挿法が画質の改善に有効であることが分かる.

さらに, 冗長ビット数を同一にした場合の代数的手法との比較の結果を図 3.4~3.7 に示す. ただし, 冗長ビット数が 0.5 ビットの場合は部分内挿法の結果のみを示している.

これらの結果より, 部分内挿法を適用すると $\varepsilon \geq 5 \times 10^{-3}$ の領域において式 (3.1) が成立していることが分かる. 代数的手法では与えることのできない 0.5 ビットの冗長を与えた時でさえ $\varepsilon \geq 5 \times 10^{-3}$ の領域で代数的手法より歪が少なくなっていることは興味深いことである.

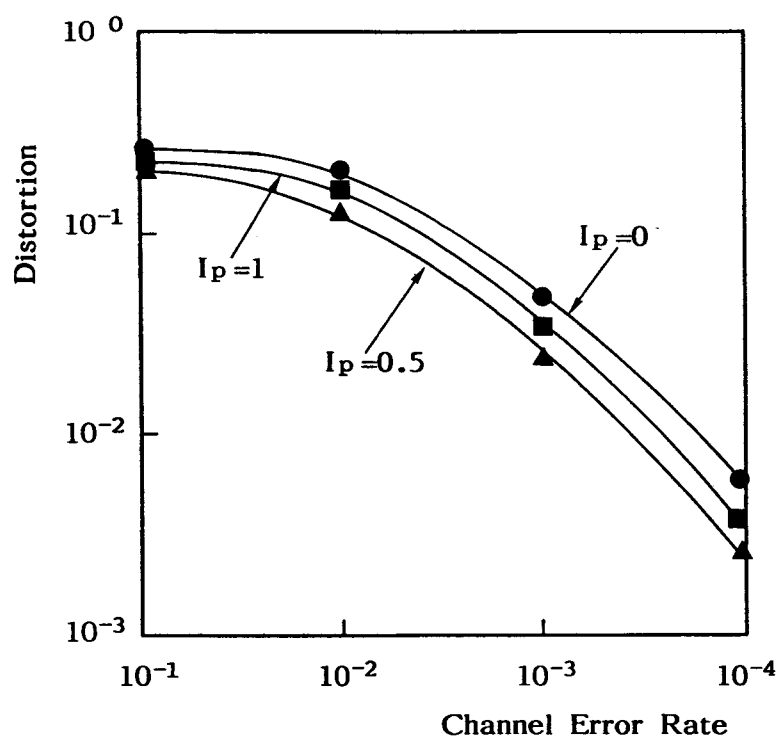


図 3.3: 部分内挿率と画面の歪

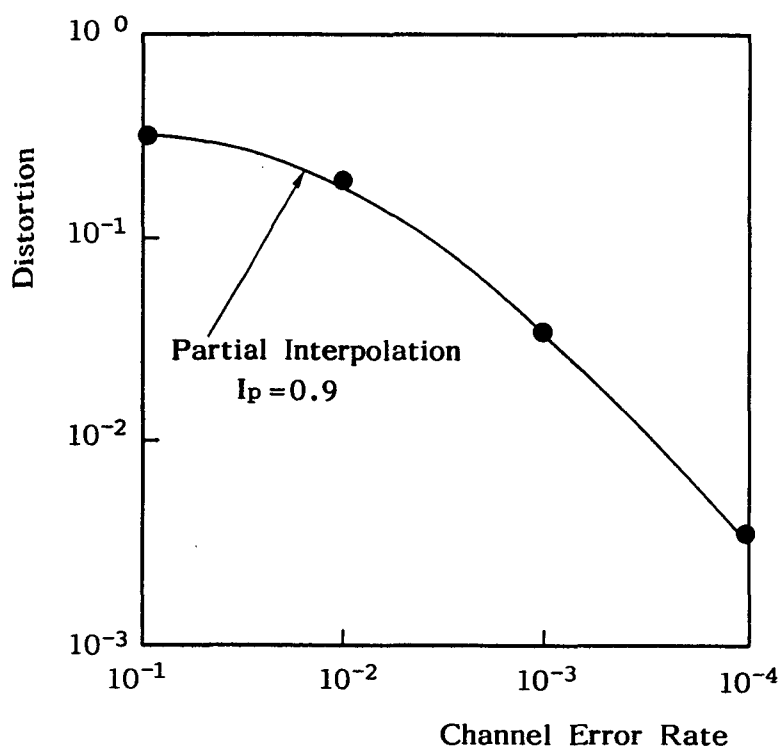


図 3.4: 通信路誤り率と画面の歪 (冗長 0.5 ビット)

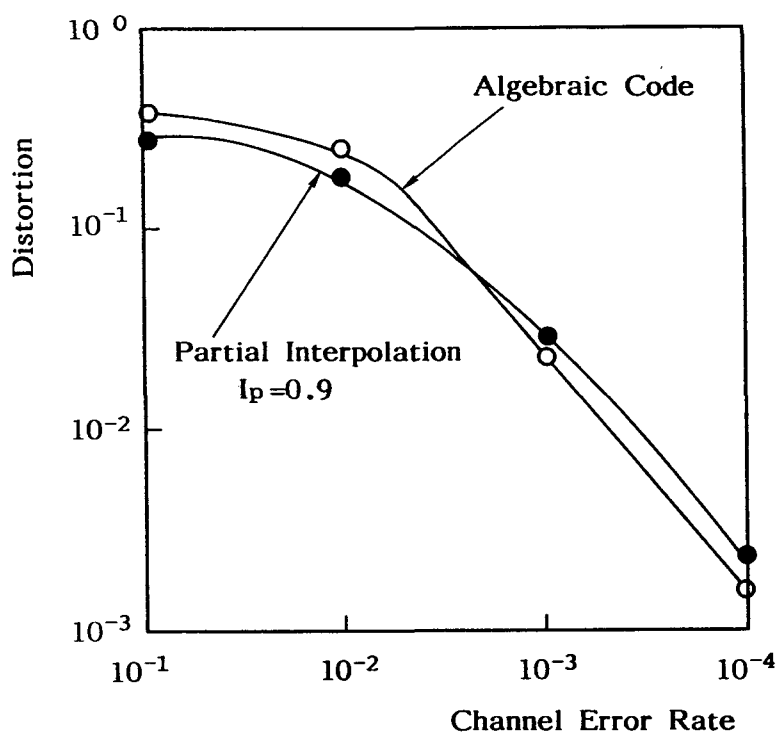


図 3.5: 通信路誤り率と画面の歪 (冗長 1 ビット)

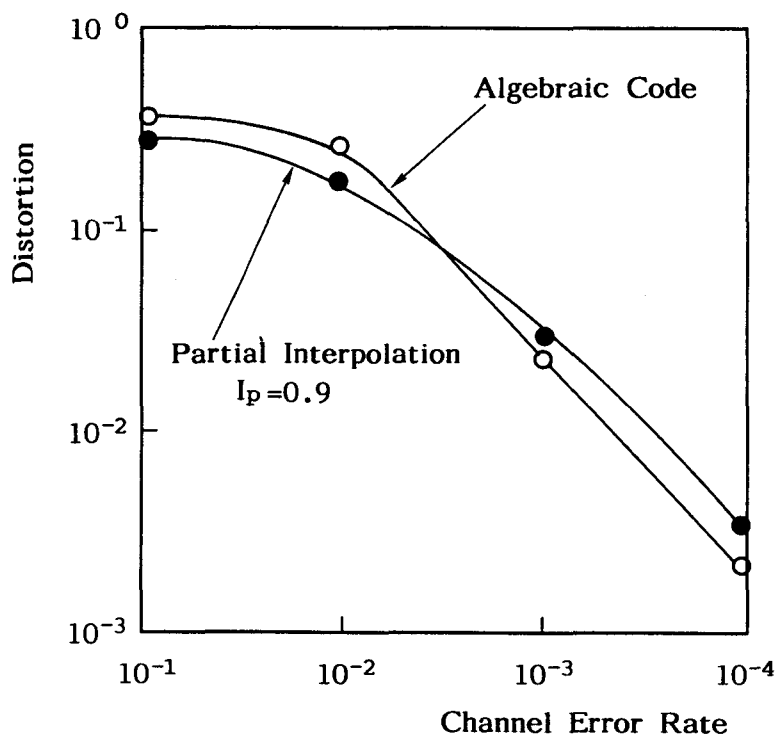


図 3.6: 通信路誤り率と画面の歪 (冗長 2 ビット)

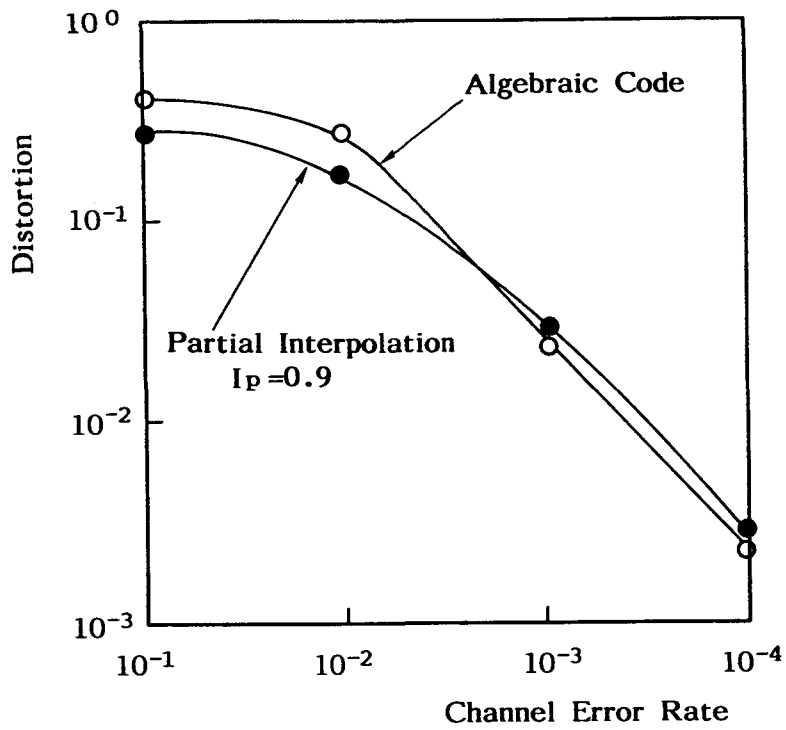


図 3.7: 通信路誤り率と画面の歪 (冗長 3 ビット)

3.4 結言

本章においては、無符号化区間挿入法を画像符号化に応用した場合について考察を行った。無符号化区間挿入法は、誤りを検出した位置の近傍までは正しい復号を行っているという性質を有しているため、誤りが検出された位置の近傍までは復号結果をそのまま用いそれ以後を直前の走査線から内挿するという部分内挿法を提案した。さらに、実際の 2 値画像を用いた計算機シミュレーションによりその有効性を確認した。その結果、通信路誤り率が比較的大きい範囲では代数的手法より小さな歪が得られそれ以外ではほぼ同等の性能が得られることを明らかにした。さらに、代数的手法のように改めて誤り検出能力を与えるための処理を施す必要がないため構成の簡単化・処理時間の短縮が図れるという利点があることを述べた。

第 4 章

分解能成分近似ブロック符号化法

4.1 緒言

本章においては，算術符号の濃淡画像符号化への一応用としてブロック符号化 (Block Truncation Coding :BTC) に基づく分解能成分近似ブロック符号化法を提案する．分解能成分近似ブロック符号化法は画像の局所的性質に基づいて符号化モードを切り替えて符号化を行う方式であるが，その符号化モード情報の符号化に算術符号を用いることによって符号化特性の向上を図っている．具体的な符号化アルゴリズムを示すと共に，実際の濃淡画像を用いた計算機シミュレーションにより広い範囲の符号化レートで従来の BTC を上回る性能が得られることを明らかにする．

4.2 ブロック符号化法

Delp ら^[50]あるいは岸本ら^[51]によって提案された BTC は符号化・復号化処理が簡単であり，比較的良好な画質が得られる符号化法として活発に研究されている．BTC はまず符号化すべき濃淡画像を $N \times N$ 画素の部分画像 (ブロック) に分割し，その後ブロック内の各画素の階調値を，複数のあらかじめ定められた階調値の内の 1 つに近似して符号化を行う手法である．BTC にお

いては各ブロックは、近似に用いられた階調値 (階調成分) とブロック内の各画素がいずれの階調値に近似されたかを示す情報 (分解能成分) の2つの成分に符号化される。BTCを基本として、復号側でフィルタ処理を施すことでブロック状の歪を減少させ視覚特性の向上を図った手法^[53]や、画像の局所的性質を考慮して階調成分の数を適応的に変化させたり^[60]、類似する性質のブロックを連結することによってブロックサイズを適応的に変化させる手法^[54]等が提案されている。さらに、BTC単独ではなく他の符号化手法と組み合わせる手法^[59]も提案されている。また、最近では濃淡画像だけでなくカラー画像の符号化への応用も試みられている^[57]。Delpらあるいは岸本らの方式は細部において異なっているがここでは岸本らの方式について述べる。

まず、符号化すべき濃淡画像を $N \times N$ 画素の部分画像 (ブロック) に分割し、ブロック内の画素の階調値を $x(i, j)$ ($i, j = 0, 1, \dots, N-1$) とする。 $x(i, j)$ から次式によってブロック内の平均階調値 $\bar{X}$ を求める。

$$\bar{X} = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \quad (4.1)$$

次に、この $\bar{X}$ をしきい値として $\bar{X}$ 以上の階調値を持つ画素の平均階調値 g_1 および $\bar{X}$ より低い階調値を持つ画素の平均階調値 g_0 を求める。すなわち、

$$g_0 = \frac{1}{N_0} \sum_{x(i, j) < \bar{X}} x(i, j) \quad (4.2)$$

$$g_1 = \frac{1}{N_1} \sum_{x(i, j) \geq \bar{X}} x(i, j) \quad (4.3)$$

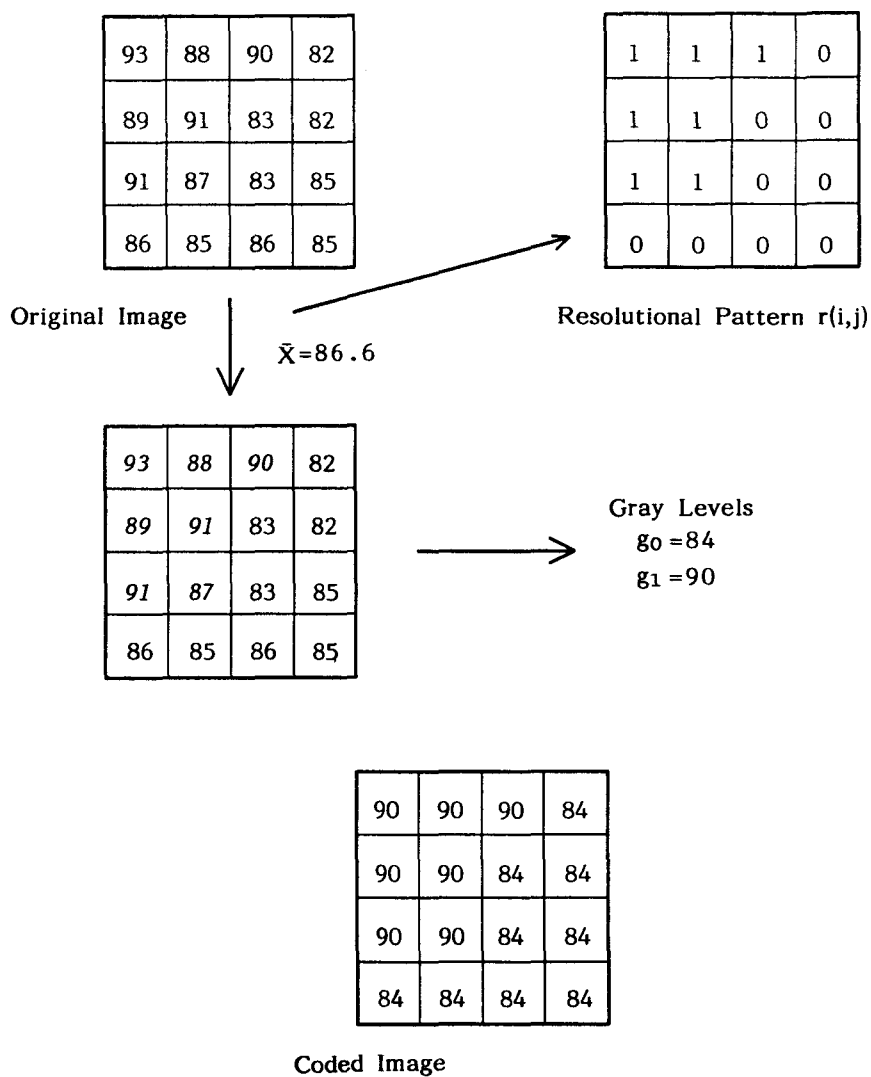
である。ただし、 N_0 は $x(i, j) < \bar{X}$ である画素数、 N_1 は $x(i, j) \geq \bar{X}$ である画素数を表す。このようにして得られる g_0 、 g_1 を階調成分と呼ぶ。ブロック内の各画素の階調値は g_0 、 g_1 のいずれかの階調値に符号化される。いずれの階調値に符号化されたかを示すために次式に従って $N \times N$ の大きさの分解能成分と呼ばれる行列が求められる。

$$r(i, j) = \begin{cases} 0 & x(i, j) < \bar{X} \\ 1 & x(i, j) \geq \bar{X} \end{cases} \quad (4.4)$$

結局、ブロック符号化では1つのブロック $x(i, j)$ ($i, j = 0, 1, \dots, N-1$) は階調成分 g_0 , g_1 および分解能成分 $r(i, j)$ ($i, j = 0, 1, \dots, N-1$) へと符号化される。

復号は、分解能成分 $r(i, j)$ を参照しそれによって各画素に g_0 , g_1 を割り当てることによって行われる。

以上に述べたように BTC において伝送される情報は階調成分と分解能成分である。実際に伝送される場合は階調成分は適当な量子化を施され、分解能成分は通常はそのまま伝送されるかあるいは Huffman 符号等によって歪なく圧縮される場合がほとんどである。しかしながら、BTC が本来歪のある符号化方式であることを考えると必ずしも分解能成分のみを歪なく圧縮する必要はなく、階調成分と分解能成分の両者の符号化に歪を許容することによって総合的に符号化特性の向上を図ることが期待できる。現在までに提案されている手法のうち、分解能成分の圧縮に歪を導入したものとしては文献 [54] –[55] が挙げられる程度であり、このことに関する検討はほとんどなされていない。そこで次節ではより積極的に分解能成分の圧縮に歪を導入した分解能成分近似ブロック符号化法と呼ぶブロックサイズ可変 BTC の一手法を提案する。本手法は平均2乗歪測度のもとで、あらかじめ選択された分解能成分パターンの中から符号化されたブロックの歪ができるだけ小さくなるようなパターンを見出し、これを用いて効率的に符号化を行う手法である。さらに、画像の局所的な性質に応じて用いる分解能成分パターンの種類を変更する等、符号化のモードを適応的に切り替えることによって符号化特性の向上を図っている。その符号化モード情報の符号化に算術符号を用いることによって一層の低ビットレート化を達成する。

図 4.1: ブロック符号化の例 ($N=4$)

4.3 分解能成分パターンを用いたブロック符号化

図 4.2 に本手法の概要を示す。図 4.2 から分かるように本手法においては初めにあらかじめ用意された分解能成分パターンを用いて符号化を行う。そこで、まずこの符号化に用いる分解能成分パターンの選択方法について述べる。

4.3.1 分解能成分パターンの選択

本手法は符号化に用いるべき分解能成分の数を制限することによって分解能成分の圧縮に歪を導入し、それによって符号化特性の向上を図っている。しかしながら、分解能成分の数を制限することによって再生画像における歪の増加が予想される。従って、符号化に用いる分解能成分パターンの選択に当たっては SN 比の面だけではなく視覚的にもできるだけ歪の少なくなるような分解能成分パターンを選択する必要がある。一般に、画像の輪郭(エッジ)を乱すような歪は SN 比の低下以上に視覚的により大きな品質低下をもたらすことが知られている [62]–[63]。従って、符号化に用いる分解能成分パターンとしては輪郭部分をできる限り歪なく符号化できるようなパターンを含んでいる必要がある。また、汎用性の面から符号化に用いる分解能成分パターンはできるだけ符号化する画像に依存しないほうが望ましい。以上のような理由から、本手法では

1. 平坦パターン
2. 基本パターン
3. 重畳パターン

と呼ぶ 3 種類の分解能成分パターンを用いることにする。

平坦パターンは分解能成分がすべて“1”であるような分解能成分パターンであり、このパターンを用いて符号化されたブロック内のすべての画素は同一の

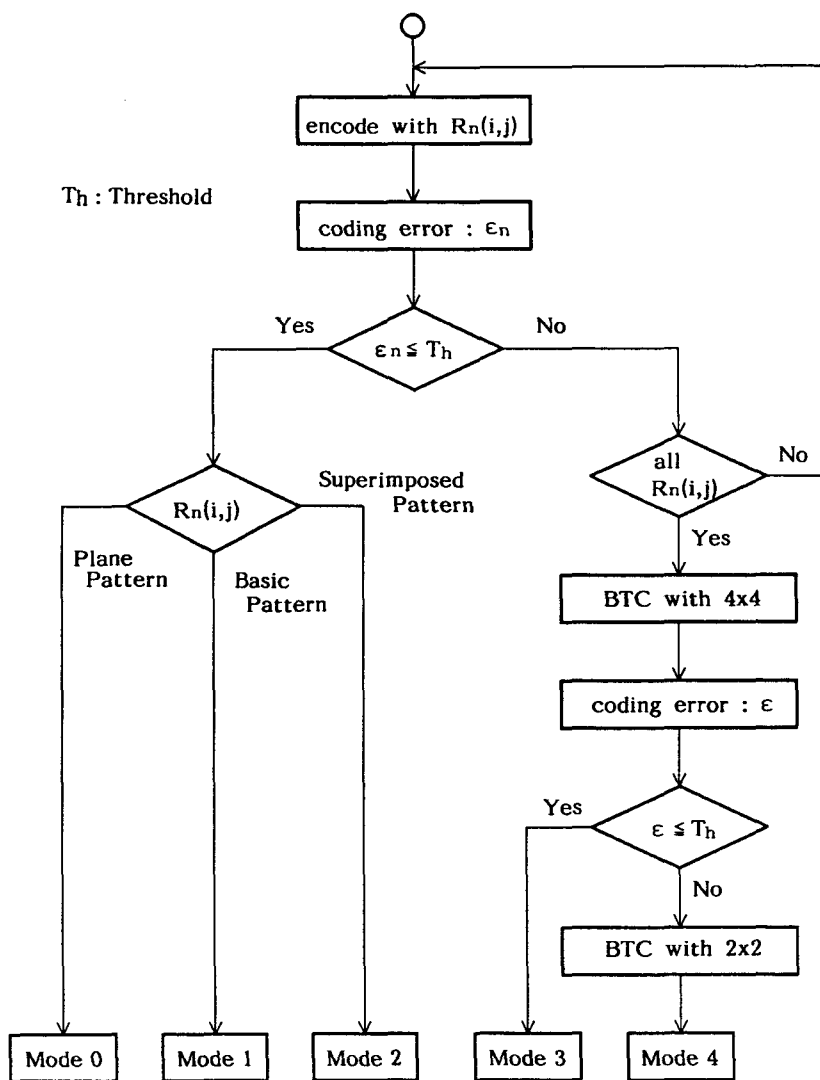


図 4.2: 符号化アルゴリズム

階調値を持つことになる。大部分の画像においては輝度変化の少ない領域の占める割合が比較的高い。そのような領域に含まれるブロックをそのまま BTC によって符号化しても、成分がすべて“1”の分解能成分と、同一の値を持つ階調成分が得られるだけであり、これを伝送することは効率的ではない。このようなブロックはブロック内の画素値を同一の値に近似してもほとんど歪は増加しないと考えられる。従って、これらのブロックは分解能成分がすべて“1”であるという情報と、1つの階調成分の値を伝送することで復号することが可能である。そのようなブロックの判定に用いられるパターンが平坦パターンである。

基本パターンは上述したように輪郭部分を含むブロックを符号化するための分解能成分パターンである。尚、本研究で考える輪郭を含むブロックとは符号化した際にブロック内が明確に2つの領域に分割されるようなブロックのことであり、3つ以上の領域に分割されるようなものは考慮しない。本手法で用いる BTC は基本的なブロックの大きさとして 4×4 画素という比較的小さいサイズを想定しているため基本パターンとして以下のような条件のパターンを用いることとする。

1. 輪郭線の方向は4種類(水平, 垂直, 両対角線方向)
2. 輪郭線の存在する位置は水平, 垂直方向については3種類, 両対角線方向については4種類

このように輪郭線の方向, 位置に関して制限を付加してもブロックの大きさが小さいため歪の増加は少なくなることが十分に期待される。本手法で用いた基本パターンを図 4.3 に示す。尚, これらのパターンを一意に決定するためには輪郭線の方向について2ビット, 輪郭線の存在する位置について2ビットの計4ビットの情報量を必要とする。

1	1	1	1

1	1	1	1
1	1	1	1

1	1	1	1
1	1	1	1
1	1	1	1

(1) Horizontal Edge

1			
1			
1			
1			

1	1		
1	1		
1	1		
1	1		

1	1	1	
1	1	1	
1	1	1	
1	1	1	

(2) Vertical Edge

1	1		
1			

1	1	1	
1	1		
1			

1	1	1	1
1	1	1	
1	1		
1			

1	1	1	1
1	1	1	1
1	1	1	
1	1		

(3) Diagonal Edge 1

		1	1
			1

	1	1	1
		1	1
			1

1	1	1	1
	1	1	1
		1	1
			1

1	1	1	1
1	1	1	1
	1	1	1
		1	1

(4) Diagonal Edge 2

Blank is "0"

図 4.3: 基本パターン

以上の平坦パターン，基本パターンだけでは実際に符号化を行う場合にパターン数の少なさから歪の増大を招く恐れがある．このため，本手法では2つの基本パターンからビット毎の論理演算によって生成されるパターンをも分解能成分パターンとして用いることによって歪の低減を図っている．これを重畳パターンと呼ぶが，重畳パターンは分解能成分パターン数の不足を補う目的で導入されたものであるので平坦パターンや基本パターンのような明確な物理的意味は有していない．論理演算としては論理積，論理和および排他的論理和が考えられるが，論理和によって生成されるパターンは“1”の数が多くなりほとんどが平坦パターンによって近似できると考えられる．事前のシミュレーションによる調査の結果からも論理和によって生成されるパターンが用いられることはほとんどないことが判明したため，ここでは論理演算としては論理積および排他的論理和の2つを考える．従って，重畳パターンを記述するためには用いられた2つの基本パターンと演算を規定しなければならないため9ビットの情報量が必要となる．

次節ではこれらの分解能成分パターンを用いた符号化法について述べよう．

4.3.2 選択された分解能成分パターンを用いた符号化

4.3.1において定めた分解能成分パターンを用いた符号化法を以下に述べる．それに先立ち以下のように記号を定義する．

$x(i, j)$	: 原画像から分割された 4×4 画素のブロック内の画素の階調値 ($i, j = 0, 1, \dots, 3$)
$y(i, j)$	: 符号化されたブロック内の画素の階調値
$R_n(i, j)$	: あらかじめ定められた分解能成分パターン
M	: $R_n(i, j)$ 内の0の数

《符号化法》

1. $R_n(i, j)$ を用いて次式によって2つの階調値 g_0 , g_1 を求める.

$$g_0 = \frac{1}{M} \sum_{R_n(i,j)=0} x(i, j) \quad (4.5)$$

$$g_1 = \frac{1}{4 \times 4 - M} \sum_{R_n(i,j)=1} x(i, j) \quad (4.6)$$

2. $R_n(i, j)$ に従い g_0 , g_1 を次のように割り当てる.

$$y(i, j) = \begin{cases} g_0 & R_n(i, j) = 0 \\ g_1 & R_n(i, j) = 1 \end{cases} \quad (4.7)$$

また, この時の符号化歪を

$$\varepsilon_n = \frac{1}{4 \times 4} \sum_{i=0}^3 \sum_{j=0}^3 \{x(i, j) - y(i, j)\}^2 \quad (4.8)$$

で定義する. 以上の符号化を ε_n がしきい値以下となるまで繰り返す.

基本パターンを用いた符号化の例を図4.4に示す.

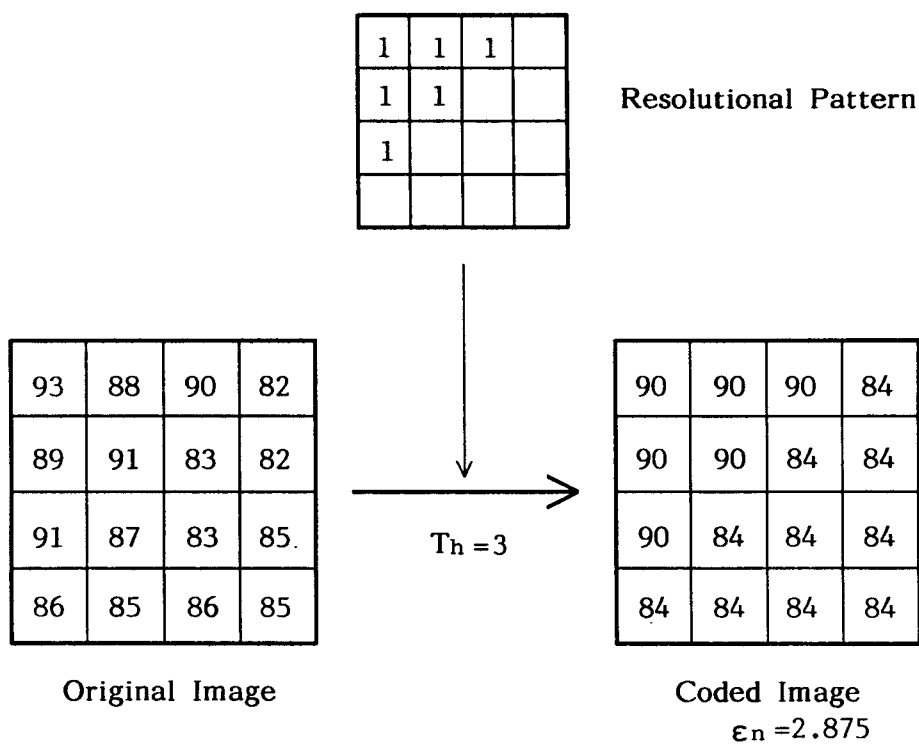


図 4.4: 基本パターンを用いた符号化例

4.3.3 符号化モード

本手法においてはしきい値によって符号化の方法が5つのモードに分類される。本節では、図4.2に示されている各符号化モードについて述べる。尚、各々のモードで出力される階調成分は4.3.4で述べるような量子化を施される。

(1) モード0

分解能成分パターンとして平坦パターンを用いて符号化されるモードであり、出力される情報はブロック内の画素の平均階調値 $\bar{X}$ のみであり、これは式(4.1)において $N = 4$ とすることによって与えられる。

(2) モード1

分解能成分パターンとして基本パターンを用いて符号化するモードであり、出力される情報は式(4.5)(4.6)から求められる g_0 , g_1 および用いられた基本パターンである。

(3) モード2

分解能成分パターンとして重畳パターンを用いて符号化を行うモードであり、出力される情報はモード1と同様にして求められる g_0 , g_1 および用いられた2つの基本パターンと演算の種類を示す情報である。

(4) モード3

4×4 画素の大きさと通常のBTCによって符号化を行うモードである。出力される情報は式(4.2)～(4.4)によって求められる階調成分 g_0 , g_1 および分解能成分 $r(i, j)$ ($i, j = 0, 1, \dots, 3$) である。

(5) モード4

4×4 画素のブロックを図4.5のように4つの 2×2 画素のサブブロックに分割してサブブロックごとに通常のBTCによって符号化を行うモードである。出力される情報は4組の階調成分 g_0 , g_1 および分解能成分 $r(i, j)$ ($i, j = 0, 1$) である。

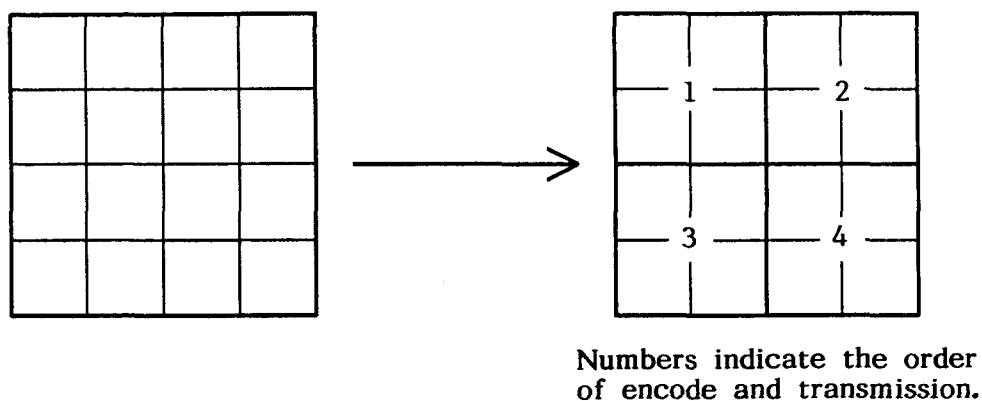


図 4.5: 2 × 2 画素への分割法

4.3.4 階調成分の符号化

モード 0 以外のモードでは符号化の結果、2つの階調成分 g_0 、 g_1 が得られる。本手法においてはこれらの階調値を以下のような形で量子化を行う。

$$g_a = \langle g_0 \rangle \quad (4.9)$$

$$g_b = \begin{cases} [g_0 - g_1] & \text{モード 1} \\ \{g_0 - g_1\} & \text{それ以外のモード} \end{cases} \quad (4.10)$$

ただし、 $\langle \cdot \rangle$ は 8 ビット線形量子化、 $[\cdot]$ は 4 ビット線形量子化、 $\{\cdot\}$ は 4 ビット非線形量子化を表す。モード 1 を用いて符号化されるブロックはブロック内に輪郭部分を含んでおり 2つの階調成分の差分の分布に比較的偏りが少ないため線形量子化を用い、他のモードの場合にはその差分の分布は 0 付近に集中する傾向があるため非線形量子化を用いている。

また、モード0で符号化されるブロックの内、特に直前のブロックもモード0で符号化されたものについては直前のブロックの階調成分と現ブロックの階調成分の差分の分布が0付近に集中するため、これを4ビット非線形量子化している。

4.3.5 算術符号によるモード識別フラグの符号化

本手法においては符号化の際に用いられたモードを識別するために1ブロック(4×4画素)毎にフラグを付加する必要がある(モード4においては実際の符号化は2×2画素のサブブロック単位で行われるがモードを識別するためのフラグはそれらのサブブロックを4つまとめた4×4画素のブロックについて付与している)。これをモード識別フラグと呼ぶが、各モード識別フラグの使用頻度に偏りがあるためこれをそのまま伝送することは効率的ではない。本手法ではこれらを算術符号を用いて符号化する。これは、第1章でも述べたように、算術符号は情報源の確率パラメータを前もって知らなくても符号化が可能であるという特長を有しているため、画像符号化に応用する場合には Huffman 符号等より適しているためである。その際の情報源モデルとしてはブロック間の相関を利用するために直前のブロックのモードによって条件づけられる1次マルコフモデルを用いている。

各モードにおいて必要とされる1ブロック当りの情報量を表4.1, 表4.2に示す。分解能成分および階調成分は符号化レートに関わらず一定でありそれらを表4.1に示す。一方、モード識別フラグに関しては符号化レートによって各符号化モードの使用頻度が異なり、必要とする情報量の変動するためそれらについては一例を表4.2に示している。これから、符号化レートが低くなるにつれて算術符号による符号化レート低減の効果が大きくなっていることが分かる。

表 4.1: 符号化に必要な情報量 (階調成分, 分解能成分)

モード	分解能成分 (ビット/ブロック)	階調成分 (ビット/ブロック)	合計
0	0	モード 0 の後のモード 0 : 4	4
		その他 : 8	8
1	用いた基本パターン : 4	$g_a : 8$ $g_b : 4$	16
2	用いた基本パターン と演算 : 9	$g_a : 8$ $g_b : 4$	21
3	分解能成分 : 16	$g_a : 8$ $g_b : 4$	28
4	分解能成分 : 16	$g_a : 8 \times 4$ $g_b : 4 \times 4$	64

表 4.2: 符号化に必要な情報量 (モード識別フラグ)

符号化レート (ビット/画素)	モード識別フラグ (ビット/ブロック)
1.67	1.68
1.46	1.49
1.33	1.41
1.04	1.18
0.70	0.86
0.60	0.75
0.47	0.59
0.34	0.37
0.26	0.05

4.4 計算機シミュレーション

本節においては、実際の濃淡画像を用いた計算機シミュレーションによって分解能成分近似ブロック符号化法の特性を評価する。シミュレーションに用いた画像はSIDBA8ビットPCM画像(512×512画素, GIRL)である。尚、復号側においては文献[53]で提案されたフィルタをモード0で符号化されたブロックに対して適用している。モード0で符号化されたブロックはブロック状の歪が生じやすいが、このフィルタ処理によってブロック状の歪を補正することができその結果、 SN 比の改善と視覚特性の向上を図ることができる。

符号化レートと SN 比の関係を他のBTCの特性と共に図4.6に示す。ただし、 SN 比は次式で定義する。

$$SNR = 10 \log_{10}(255^2/MSE) \quad (4.11)$$

ただし、 MSE は画素当りの平均2乗誤差を表す。

図4.6から明らかなように本手法はBTC単独の手法としては広い範囲の符号化レートにおいて優れた特性を示していることが分かる。文献[53]の方式と比較して等符号化レートで2.5～3.5(dB)程度、文献[54]の方式に対しては1.7～2.3(dB)程度の SN 比の改善が得られている。文献[54]の方式は実際の符号化を行う前に性質の類似するブロックを統合するという処理が必要であり、この処理が比較的複雑であることを考慮すると、本手法のほうがより簡単であり実用的な方法であると言えよう。また、文献[60]の方式と比較した場合には、等レートで0.3～1.0(dB)程度の改善が得られている。特に、1(ビット/画素)以下の領域における改善の度合いが大きくなっていることは注目に値しよう。この領域ではブロック全体の85%以上が平坦パターン、基本パターンおよび重畳パターンを用いて符号化されており、本研究で導入した3種類の分解能成分パターンが分解能成分の圧縮に大きく寄与していることが分かる。

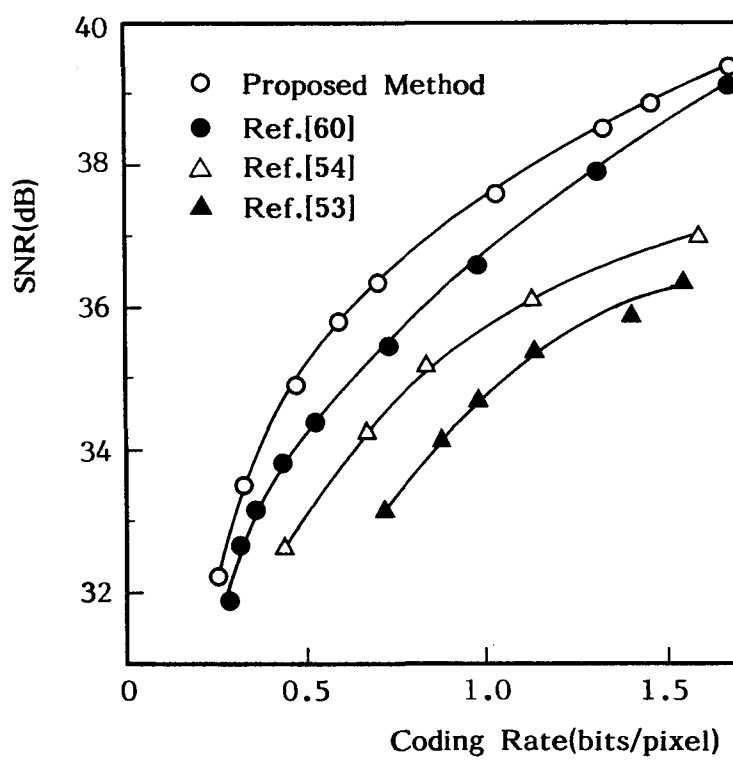


図 4.6: 符号化特性

4.5 結言

本章においては、あらかじめ選択した分解能成分パターンの中から符号化されたブロックにおける平均2乗歪ができる限り小さくなるようなパターンを用いて符号化を行うことによって総合的に符号化特性の向上を図った分解能成分近似ブロック符号化と呼ぶ新たな符号化法を提案した。本方式は符号化モード識別フラグの符号化に算術符号を用いており実際の濃淡画像を用いた計算機シミュレーションの結果、従来方式に比較して等符号化レートにおいて SN 比で $0.3 \sim 3.5(\text{dB})$ 程度改善が得られた。また、算術符号を使用することにより符号化レートが低くなるにつれてより低ビットレート化が図れることが明らかとなった。

第 5 章

結論

本論文は、算術符号とその画像符号化への応用に関して筆者が行った研究の成果をまとめたものである。本章では、本研究により得られた成果について総括する。

第 2 章では、算術符号の欠点である誤り伝播の問題に対して一つの解決法を与えるものとして、符号語に生起した誤りを検出する能力を有する無符号化区間挿入法と呼ぶ新たな算術符号の構成法を提案した。Pasco のアルゴリズムを例にとり具体的な符号化・復号化アルゴリズムを示し、その平均符号長ならびに誤り検出確率の評価式の導出を行った。無符号化区間挿入法は従来の算術符号化を一般化することによって実現しており、算術符号の特長を損なうことなく誤り検出能力を付与することが可能な方式である。さらに通常の通信路符号化と組み合わせることにより自由度に富んだ冗長および誤り検出能力を与えることができることを計算機シミュレーションによって明らかにした。

第 3 章では、第 2 章で提案した無符号化区間挿入法を画像符号化に応用した場合について考察を行った。無符号化区間挿入法は、誤りを検出した位置の近傍までは正しい復号を行っているという性質を有しているため、誤りが検出された位置の近傍までは復号結果をそのまま用いそれ以後を直前の走査線から

内挿するという部分内挿法を提案した。さらに，部分内挿法を用いることにより従来方式とほぼ同等の画質を確保することが可能であることを実際の 2 値画像を用いて確認した。また，無符号化区間挿入法では符号化の過程で誤り検出能力を付与していることから通常の場合のように改めて誤り検出能力を付加する処理を施す必要がなく構成の簡単化・処理時間の短縮が図れることを述べた。

第 4 章では，算術符号の濃淡画像符号化への一応用として分解能成分近似ブロック符号化と呼ぶ新たなブロック符号化法の提案を行った。本方式は，画像の局所的な性質によって符号化モードを適宜選択して符号化効率を高めるものである。その際，人間の視覚特性等を基準としてあらかじめ選択された分解能成分を用いることによって分解能成分の圧縮を図っている。さらに，符号化モード情報の符号化に算術符号を用いることによってより一層の低ビットレート化を図ることが可能となった。実際の濃淡画像を用いて計算機シミュレーションを行った結果，従来方式と比較して SN 比 - 符号化レート特性の改善が得られることを明らかにした。

謝辞

本研究は、大阪大学工学部通信工学科教授滑川敏彦博士(現 名誉教授)ならびに大阪大学工学部通信工学科教授森永規彦博士の御指導のもとに、筆者が大阪大学大学院工学研究科在学中に行った研究の成果をまとめたものである。本研究を遂行するにあたり、両先生から賜った懇意なる御教示、御鞭撻に対して衷心より感謝の意を表する次第である。

さらに、筆者は大阪大学工学部通信工学科教授手塚慶一博士より、有益な御教示、御助言を賜った。ここに感謝の意を表する。

また、筆者の大学院在学中、講義等を通じて通信工学全般に渡る御指導を賜った大阪大学教授熊谷信昭博士(現 総長)、同教授中西義郎博士(現 名誉教授)、同教授倉園貞夫博士、同教授北橋忠宏博士を始めとする諸先生方に対して厚く感謝を申し上げる。

本研究の途上、筆者は京都工芸繊維大学工芸学部教授笠原正雄博士より、昼夜を分かたぬ御激励ならびに懇切丁寧なる御指導を賜った。ここに深謝申し上げる。

本研究の遂行にあたり、親身なる御助言、御激励を賜った大阪大学工学部助教授村田正博士、同助手佐藤正志博士、同助手塚本勝俊氏並びに大阪大学大型計算機センター講師片山正昭博士に厚く御礼申し上げる。

また、熱心な御討論、御助言を頂いた大阪大学工学部通信工学科森永研究

室の諸氏に厚く御礼申し上げる。

最後に、筆者の大学院在学中、公私に渡り好意溢れる御助言、御激励ならびに御厚情を賜った神戸大学工学部講師常盤欣一郎博士ならびに姫路獨協大学一般教育部助手松田泰至博士に心からの感謝を申し上げます。

参考文献

- [1] C.E.Shannon, "A Mathematical Theory of Communication, " *B.S.T.J.*, vol.27, pp.379–423, pp.623–656, July, Oct. 1948.
- [2] D.A.Huffman, "A Method for the Construction of Minimum Redundancy Codes, " *Proc. IRE*, vol.40, pp.1098–1101, Sept. 1952.
- [3] E.R.Kretzmer, "Statics of Television Signals, " *B.S.T.J.*, vol.31, pp.751–763, July 1952.
- [4] S.E.Levinson, L.R.Rabiner and M.M.Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition, " *B.S.T.J.*, vol.62, 4, pp.1035–1074, April 1983.
- [5] L.R.Rabiner, S.E.Levinson and M.M.Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition, " *B.S.T.J.*, vol.62, 4, pp.1075–1105, April 1983.
- [6] L.R.Rabiner and B.H.Juang, "An Introduction to Hidden Markov

- Models, " *IEEE ASSP Magazine*, pp.4-16, Jan. 1986.
- [7] L.R.Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, " *Proc. IEEE*, vol.77, 2, pp.257-286, Feb. 1989.
- [8] 荻原 春生, "隠れマルコフモデルによる非定常情報源の符号化, " 第10回情報理論とその応用シンポジウム資料, pp.591-596 (1987-11)
- [9] 森田, 小林, "線形従属過程のデータ圧縮に関する考察, " 第10回情報理論とその応用シンポジウム資料, pp.597-603 (1987-11)
- [10] 中井, 笠原, "隠れマルコフモデルの画像符号化への適用に関する一考察, " 信学技報, IT88-71, pp.15-18 (1988-11)
- [11] A.Gersho, "Asymptotically Optimal Block Quantization, " *IEEE Trans. Inform. Theory*, vol. IT-25, pp.373-380, July 1979.
- [12] Y.Linde, A.Buzo and R.M.Gray, "An Algorithm for Vector Quantizer Design, " *IEEE Trans. Commun.*, vol.COM-28, pp.84-95, Jan. 1980.
- [13] R.M.Gray, "Vector Quantization, " *IEEE ASSP Magazine*, pp.4-29, April 1984.
- [14] M.Goldberg and H.Sun, "Image Sequence Coding Using Vector Quantization, " *IEEE Trans. Commun.*, vol.COM-34, pp.703-710, July 1986.

- [15] N.Ahmed, T.Narajan and K.R.Rao, "On Image Processing and a Discrete Cosine Transform, " *IEEE Trans. Computer*, vol.C-23, pp.90-93, 1974.
- [16] A.Rosenfeld and A.C.Kak, *Digital Picture Processing (Volume 1)*, Academic Press, New York, 1982.
- [17] ノーマン・アブラムソン著, 宮川 洋訳, "情報理論入門, " 好学社, 1981.
- [18] R.C.Pasco, "Source Coding Algorithms for Fast Data Compression, " Ph.D. dissertation for Department of Electrical Engineering, Stanford University, CA, 1976.
- [19] J.J.Rissanen, "Generalized Kraft Inequality and Arithmetic Coding, " *IBM Res. Develop.*, vol.20, pp.198-203, May 1976.
- [20] F.Rubin, "Arithmetic Stream Coding Using Fixed Precision Registers, " *IEEE Trans. Inform. Theory*, vol.IT-25, pp.672-675, Nov. 1979.
- [21] M.Guazzo, "A General Minimum-Redundancy Source-Coding Algorithm, " *IEEE Trans. Inform. Theory*, vol.IT-26, pp.15-25, Jan. 1980.
- [22] C.B.Jones, "An Efficient Coding System for Long Source Sequences, " *IEEE Trans. Inform. Theory*, vol.IT-27, pp.280-291, May 1981.
- [23] G.G.Langdon, Jr. and J.Rissanen, "A Simple General Binary Source Code, " *IEEE Trans. Inform. Theory*, vol.IT-28, pp.800-803, Sept. 1982.

- [24] G.G.Langdon, Jr., "Compression of Black-White Images with Arithmetic Coding, " *IEEE Trans. Commun.*, vol.COM-29, pp.858-867, Jan. 1981.
- [25] 鳥山裕史, "算術符号を用いた二値画像の階層的符号化方式, " 信学技報, IE86-63, pp.17-24 (1986-10)
- [26] 加藤, 安田, "算術符号による中間調画像の高能率符号化, " 信学技報, IE82-57, pp.17-22 (1982-10)
- [27] 北田, 森田, 有本, "算術符号を用いたデジタル濃淡画像のデータ圧縮, " 情報理論とその応用研究会第7回シンポジウム資料, pp.459-464 (1984-11)
- [28] 伊勢田, 斉藤, "算術符号を用いた音声信号の適応予測符号化方式, " 信学論 (A), J68-A, 11, pp.1253-1260 (1985-11)
- [29] J.Rissanen and G.G.Langdon, Jr., "Universal Modeling and Coding, " *IEEE Trans. Inform. Theory*, vol.IT-27, pp.12-23, Jan. 1981.
- [30] J.Rissanen, "A Universal Data Compression System, " *IEEE Trans. Inform. Theory*, vol.IT-29, pp.656-664, Sep. 1983.
- [31] J.Rissanen, "Universal Coding, Information, Prediction, and Estimation, " *IEEE Trans. Inform. Theory*, vol.IT-30, pp.629-636, July 1984.
- [32] J.Rissanen and G.G.Langdon, Jr., "Arithmetic Coding, " *IBM Res. Develop.*, vol.23, pp.149-162, Mar. 1979.

- [33] 森田, 北田, 有本, “算術符号化方式の符号化効率について,” 情報理論とその応用研究会第5回シンポジウム資料, pp.111-118 (1982-11)
- [34] 韓 太舜, “データ圧縮理論の最近の進歩 [1], ” 信学誌, 67, 2, pp.185-191 (1984-2)
- [35] G.G.Langdon, Jr., “An Introduction to Arithmetic Coding,” *IBM Res. Develop.*, vol.28, pp.135-149, Mar. 1984.
- [36] 鈴木, 佐藤, 小林, “任意の出力分布を実現する算術符号,” 情報理論とその応用研究会第7回シンポジウム資料, pp.580-585 (1984-11)
- [37] 川端 勉, “マルコフ情報源に対する VF 算術符号の情報圧縮率の解析,” 第9回情報理論とその応用シンポジウム資料, pp.231-236 (1986-10)
- [38] 島田, 中村, “算術符号の桁上がり制御法,” 第9回情報理論とその応用シンポジウム資料, pp.237-242 (1986-10)
- [39] J.Rissanen and K.M.Mohiuddin, “A Multiplication-Free Multialphabet Arithmetic Code,” *IEEE Trans. Commun.*, vol.COM-37, pp.93-98, Feb. 1989.
- [40] 鈴木, 小林, “拡張された VV 算術符号,” 情報理論とその応用研究会第6回シンポジウム資料, pp.65-70 (1983-10)
- [41] 中井, 笠原, “誤り検出機能を付加した算術符号化とその応用,” 信学論 (D-II), J72-D-II, 4, pp.568-576 (1989-4)
- [42] 中井, 笠原, 滑川, “誤り検出機能を持たせた情報源符号化に関する二・三の考察,” 信学技報, IT85-17, pp.25-30 (1985-7)

- [43] 中井, 笠原, 滑川, “算術符号を用いた画像符号化に関する考察,” 信学技報, CS85-133, pp.1-4 (1986-1)
- [44] 中井, 笠原, 滑川, “誤り検出機能を持たせた算術符号化について,” 昭和 61 年度電子通信学会総合全国大会 1425 (1986)
- [45] 中井, 笠原, “算術符号を応用した画像情報の圧縮と誤り制御について,” 昭和 61 年電気関係学会関西支部連合大会 S12-9 (1986)
- [46] 中井, 笠原, 滑川, “誤り検出算術符号のパフォーマンスについて,” 昭和 62 年度電子通信学会情報・システム部門全国大会 7 (1987)
- [47] 中井, 笠原, “算術符号の画像符号化への応用について,” 1988 年度画像符号化シンポジウム 4-2 (1988-9)
- [48] P.G.Neumann, “Efficient Error-Limiting Variable-Length Codes,” *IRE Trans. Inform.*, vol. IT-8, pp.292-304, July 1962.
- [49] J.C.Maxted and J.P.Robinson, “Error Recovery for Variable Length Codes,” *IEEE Trans. Inform. Theory*, vol.IT-31, pp.794-801, Nov. 1985.
- [50] E.J.Delp and O.R.Mitchell, “Image Compression Using Block Truncation Coding,” *IEEE Trans. Commun.*, vol. COM-27, 9, pp.1335-1342, Sept. 1979.
- [51] 岸本, 三ツ矢, 星田, 釜江, “静止画像のブロック符号化法,” 信学論(B), J62-B, 1, pp.1-9 (1979-1)

- [52] 小林 誠, “多階調ファクシミリ信号の差分ブロック符号化方式,” 信学論 (B), J66-B, 2, pp.177-184 (1983-2)
- [53] 三ツ矢, 保坂, 末永, “後置フィルタ処理によるブロック符号化画像の画質改善法の検討,” 信学技報, IE84-46, pp.25-32 (1984-7)
- [54] 中村, 小松, 三王, 南, “ブロック統合処理に基づく可変ブロックサイズ符号化,” 信学論 (B), J69-B, 1, pp.1111-1120 (1986-10)
- [55] V.R.Udpikar and J.P.Raina, “BTC Image Coding Using Vector Quantization,” *IEEE Trans. Commun.*, vol. COM-35, 3, pp.352-356, Mar. 1987.
- [56] 越智, 小林, “濃淡画像の差分ブロック符号化方式,” 信学論 (B), J70-B, 1, pp.66-76 (1987-1)
- [57] 三王, 中村, 南, “カラー画像の多段階ブロック符号化,” 信学技報, IE87-71, pp.19-24 (1987-10)
- [58] 石橋 聡, “静止画像のブロック符号化における符号化パラメータの最適化,” 信学論 (B), J71-B, 10, pp.1102-1109 (1988-10)
- [59] 伊東, 木原, 嶋田, 山口, 宇都宮, “画像の階調数選択形ブロック符号化方式,” 信学論 (B), J71-B, 2, pp.1528-1536 (1988-12)
- [60] 嶋田, 伊東, 宇都宮, “静止画像の階調数選択形ブロック符号化方式,” 信学技報, CS86-141, pp.39-46 (1987-2)
- [61] 堀田, 宮原, “輪郭検出を用いた適応型ブロック符号化,” 1989年テレビジョン学会全国大会 19-7 (1989-7)

- [62] M.Miyahara and K.Kotani, "Block Distortion in Orthogonal Transform Coding — Analysis, Minimization, and Distortion Measure, " *IEEE Trans. Commun.*, vol.COM-33, pp.90-96, Jan. 1985.
- [63] 宮原 誠, "時空間相関の視・知覚特性とテレビ画像の高品質化に重要な物理要因 — 自然さ, 滑らかさ, 連続性, 安定さについて —, " *テレビ誌*, vol.40, 1, pp.46-53 (1986-1)

付録 A

無符号化区間挿入法を用いた場合の平均符号長の導出

Z を $0 < Z < 1$ を満たす実数とし, T を Z の 2 進数展開したものを有効桁数 m 桁で打ち切った数とする時,

$$\frac{T}{Z} > 1 - 2^{1-m} \quad (\text{A.1})$$

が成立する. 符号化アルゴリズム 2. の手続きを i 回行った後の各パラメータの値を添字 i を付して表すものとする, 上式および符号化アルゴリズムから

$$T_{i-1}C_i(x_i)(1 - 2^{1-m}) < T_i \leq T_{i-1}C_i(x_i) \quad (\text{A.2})$$

が成立する. ここで,

$$\ell_i = \log T_{i-1} - \log T_i \quad (\text{A.3})$$

$$V_2(m) = -\log(1 - 2^{1-m}) \quad (\text{A.4})$$

とし, 式 (A.2) の対数をとると,

$$-\log C_i(x_i) \leq \ell_i < -\log(C_i(x_i)) + V_2(m) \quad (\text{A.5})$$

となる. 一方, 長さ i の系列を $X(i)$ で表すとする

$$C_i(x_i) = k(x_i)P(x_i|X(i-1)) \quad (\text{A.6})$$

であるので、上式を代入すると

$$\begin{aligned} -\log k(x_i) - \log P(x_i|X(i-1)) &\leq \ell_i \\ &< -\log k(x_i) - \log P(x_i|X(i-1)) + V_2(m) \end{aligned} \quad (\text{A.7})$$

となる。すべての i について和をとると

$$\begin{aligned} -\sum_{i=1}^N \log k(x_i) - \log P(X(N)) &\leq \sum_{i=1}^N \ell_i \\ &< -\sum_{i=1}^N \log k(x_i) - \log P(X(N)) + NV_2(m) \end{aligned} \quad (\text{A.8})$$

となる。 $\lceil x \rceil$ を x を超える最小の整数を表すものとする符号長 L は、

$$L = \left\lceil \sum_{i=1}^N \ell_i \right\rceil + 1 \quad (\text{A.9})$$

で与えられるので、

$$\sum_{i=1}^N \ell_i + 1 \leq L < \sum_{i=1}^N \ell_i + 2 \quad (\text{A.10})$$

である。上式と式 (A.8) より

$$\begin{aligned} -\sum_{i=1}^N \log k(x_i) - \log P(X(N)) + 1 &\leq L \\ &< -\sum_{i=1}^N \log k(x_i) - \log P(X(N)) + NV_2(m) + 2 \end{aligned} \quad (\text{A.11})$$

を得る。これをすべての $X(N)$ について平均して、平均符号長 L_{av} の評価式として次式を得る。

$$H_P + 1 + \Delta \leq L_{av} < H_P + NV_2(m) + 2 + \Delta \quad (\text{A.12})$$

ただし,

$$\Delta = - \sum_{X(N)} P(X(N)) \sum_{i=1}^N \log k(x_i | x_{i-1}) \quad (\text{A.13})$$

$$V_2(m) = -\log(1 - 2^{1-m}) \quad (\text{A.14})$$

であり, H_P は情報源のエントロピー, $P(X(N))$ は系列 $X(N)$ の生起確率である.

これを, 実用上重要な無記憶2値情報源の場合について求めてみると次のようになる.

$$NH_{P_2} + 1 + \Delta \leq L_{av} < NH_{P_2} + NV_2(m) + 2 + \Delta \quad (\text{A.15})$$

ただし,

$$\Delta \approx -N\{P(0) \log k(0) + P(1) \log k(1)\} \quad (\text{A.16})$$

$$H_{P_2} = -P(0) \log P(0) - P(1) \log P(1) \quad (\text{A.17})$$

H_{P_2} は無記憶2値情報源のエントロピーである.