

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3989999号  
(P3989999)

(45) 発行日 平成19年10月10日(2007.10.10)

(24) 登録日 平成19年7月27日(2007.7.27)

(51) Int. Cl.	F I				
<b>HO4N</b>	<b>1/41</b>	<b>(2006.01)</b>	HO4N	1/41	B
<b>GO6T</b>	<b>1/20</b>	<b>(2006.01)</b>	GO6T	1/20	Z
<b>HO3M</b>	<b>7/30</b>	<b>(2006.01)</b>	HO3M	7/30	A
<b>HO4N</b>	<b>1/413</b>	<b>(2006.01)</b>	HO4N	1/413	D

請求項の数 14 (全 74 頁)

(21) 出願番号	特願平9-109790	(73) 特許権者	000006747
(22) 出願日	平成9年4月25日(1997.4.25)		株式会社リコー
(65) 公開番号	特開平10-84484		東京都大田区中馬込1丁目3番6号
(43) 公開日	平成10年3月31日(1998.3.31)	(74) 代理人	100073760
審査請求日	平成14年3月19日(2002.3.19)		弁理士 鈴木 誠
審査番号	不服2005-10046(P2005-10046/J1)	(72) 発明者	アレクサンダー エフ キース
審査請求日	平成17年5月26日(2005.5.26)		アメリカ合衆国 カリフォルニア州 94
(31) 優先権主張番号	08/642518		025 メンローパーク サンド ヒル
(32) 優先日	平成8年5月3日(1996.5.3)		ロード 2882 リコーコーポレーシ
(33) 優先権主張国	米国(US)		ン内
		(72) 発明者	エドワード エル シュワルツ
			アメリカ合衆国 カリフォルニア州 94
			025 メンローパーク サンド ヒル
			ロード 2882 リコーコーポレーシ
			ン内

最終頁に続く

(54) 【発明の名称】 データ圧縮システム

(57) 【特許請求の範囲】

【請求項1】

入力データに可逆ウェーブレット変換を実行して、前記入力データの多重解像度分解を表す係数の系列を生成する可逆ウェーブレット変換部と、該可逆ウェーブレット変換部に接続され、前記係数をビットプレーン順の視覚的に重要な順に順序付けることで埋め込みデータストリームを出力する埋め込み順序付け量子化器と、該埋め込み順序付け量子化器に接続され、前記埋め込みデータストリーム中のデータを重要性に基づきモデル化するコンテキストモデルとからなるウェーブレット方式のコード、

入力データをバイナリ符号化スキームにより符号化するバイナリ方式のコード、

前記ウェーブレット方式のコードのコンテキストモデルと前記バイナリ方式のコードとに共通に接続され、前記ウェーブレット方式のコードと前記バイナリ方式のコードに共用されるエントロピー・コード、

及び、前記ウェーブレット方式のコード又は前記バイナリ方式のコードを選択する選択制御部、

を有することを特徴とするデータ圧縮システム。

【請求項2】

請求項1記載のデータ圧縮システムにおいて、前記バイナリ方式のコードはグレイ符号化を実行することを特徴とするデータ圧縮システム。

【請求項3】

請求項1記載のデータ圧縮システムにおいて、前記エントロピー・コードは有限状態マ

10

20

シン・コードからなることを特徴とするデータ圧縮システム。

【請求項 4】

請求項 3 記載のデータ圧縮システムにおいて、前記有限状態マシン・コードはルックアップテーブルからなることを特徴とするデータ圧縮システム。

【請求項 5】

請求項 1 記載のデータ圧縮システムにおいて、前記エントロピー・コードは Q コードからなることを特徴とするデータ圧縮システム。

【請求項 6】

請求項 1 記載のデータ圧縮システムにおいて、前記エントロピー・コードは Q M コードからなることを特徴とするデータ圧縮システム。

10

【請求項 7】

請求項 1 記載のデータ圧縮システムにおいて、前記エントロピー・コードは並列コードからなることを特徴とするデータ圧縮システム。

【請求項 8】

入力データを増加関数に基づき画像のとり値に写像するヒストグラム圧縮機構、  
前記ヒストグラム圧縮機構に接続され、入力データに可逆ウェーブレット変換を実行して、前記入力データの多重解像度分解を表す係数の系列を生成する可逆ウェーブレット変換部と、該可逆ウェーブレット変換部に接続され、前記係数をビットプレーン順の視覚的に重要な順に順序付けることで埋め込みデータストリームを出力する埋め込み順序付け量子化器と、該埋め込み順序付け量子化器に接続され、前記埋め込みデータストリーム中のデータを重要性に基づきモデル化するコンテキストモデルとからなるウェーブレット方式のコード、

20

入力データをバイナリ符号化スキームにより符号化するバイナリ方式のコード、  
前記ウェーブレット方式のコードのコンテキストモデルと前記バイナリ方式のコードとに共通に接続され、前記ウェーブレット方式のコードと前記バイナリ方式のコードに共用されるエントロピー・コード、

及び、前記ウェーブレット方式のコード又は前記バイナリ方式のコードを選択する選択制御部、

を有することを特徴とするデータ圧縮システム。

【請求項 9】

請求項 8 記載のシステムにおいて、前記ヒストグラム圧縮機構はブーリアンヒストグラムを作ることを特徴とするデータ圧縮システム。

30

【請求項 10】

請求項 8 記載のデータ圧縮システムにおいて、前記ヒストグラム圧縮機構は整数値を画像データ中の全ての可能な画素値に写像することを特徴とするデータ圧縮システム。

【請求項 11】

前記ヒストグラム圧縮機構で利用されたマッピングを復号化器に通知するために接続された通知機構をさらに含むことを特徴する請求項 8 記載のデータ圧縮システム。

【請求項 12】

請求項 11 記載のデータ圧縮システムにおいて、前記マッピングは復号化器に受け取られた圧縮データに含まれるヘッダで通知されることを特徴とするデータ圧縮システム。

40

【請求項 13】

請求項 11 記載のデータ圧縮システムにおいて、ヘッダ中の 1 つのビットが、それがセットされたときに、カレント・タイルのために別のヒストグラムが利用されることを復号化器に指示することを特徴とするデータ圧縮システム。

【請求項 14】

請求項 11 記載のデータ圧縮システムにおいて、前記復号化器は、値のダイナミックレンジと等しい、ある数のビットを送ることによって通知され、該ある数のビット中の各ビットはダイナミックレンジ内の対応した値が使われるときにセットされることを特徴とするデータ圧縮システム。

50

## 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明はデータ圧縮及び伸長システムの分野に係り、特に、圧縮／伸長システムにおけるデータの非損失性（lossless）及び損失性（lossy）の符号化及び復号化の方法と装置に関する。

## 【0002】

## 【従来の技術】

データ圧縮は、大量のデータの蓄積及び伝送のために非常に有用なツールである。例えば、文書のファクシミリ伝送のような画像伝送に要する時間は、圧縮を利用して画像再生に必要とされるビット数を減らすと飛躍的に短縮される。

10

## 【0003】

従来より、多くの様々なデータ圧縮手法が存在している。圧縮手法は、おおまかに分類すると2つのカテゴリー、つまり損失性符号化と非損失性符号化とに分けることができる。損失性符号化とは、情報の損失を生じ、したがって元のデータの完全な再現が保証されない符号化のことである。損失性符号化の目標とするところは、元のデータから変わったとしても、その変化が不快であったり目だったりしないようにすることである。非損失性圧縮では、情報がすべて保存され、データは完全な復元が可能な方法で圧縮される。

## 【0004】

非損失性圧縮では、入力シンボルもしくは輝度データが出力符号語に変換される。入力としては、画像データ、音声データ、1次元データ（例えば空間的または時間的に変化するデータ）、2次元データ（例えば2つの空間軸方向に変化する（または1つの空間次元と1つの時間次元で変化する）データ）、あるいは多次元／マルチスペクトルのデータがある。圧縮がうまくいけば、その符号語は、符号化前の入力シンボル（または輝度データ）のために必要とされたビット数より少ないビット数で表現される。非損失性符号化法には、辞書符号化方式（例えば、Lempel-Ziv方式）、ランレングス符号化方式、計数符号化方式、エントロピー符号化方式がある。非損失性の画像圧縮では、圧縮は予測またはコンテキストと符号化に基づいている。ファクシミリ圧縮用JBIG規格と、連続階調画像用のDPCM（差分パルス符号変調-JPEG規格のオプション）は画像用の非損失性圧縮の例である。損失性圧縮では、入力シンボルまたは輝度データは、量子化されてから出力符号語へ変換される。量子化は、データの重要な特徴量を保存する一方、重要でない特徴量を除去することを目的としている。損失性圧縮システムは、量子化に先立ち、エネルギー集中をするための変換を利用することが多い。JPEGは画像データ用の損失性符号化法の一例である。

20

30

## 【0005】

画像信号処理における近年の開発は、効率的かつ高精度のデータ圧縮符号化方式を追求することに関心を集中してきた。変換またはピラミッド信号処理の様々な方式が提案されており、その中に多重解像度ピラミッド処理方式とウェーブレット（wavelet）ピラミッド処理方式とがある。これら2方式はサブバンド処理方式及び階層処理方式とも呼ばれる。画像データのウェーブレット・ピラミッド処理方式は、直交ミラーフィルタ（QMF）を用いてオリジナル画像のサブバンド分解をする特殊な多重解像度ピラミッド処理方式である。他の非QMFウェーブレット方式もある。ウェーブレット処理方式に関し、これ以上の情報を得るにはAntonini, M., et al., "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1, No. 2, April 1992、及びShapiro, J., "An Embedded Hierarchical Image Coder Using Zero Trees of Wavelet Coefficients", Proc. IEEE Data Compression Conference, pgs. 214-223, 1993を参照されたい。また、可逆変換に関する情報を得るには、Said, A. and Pearlman, W. "Reversible Image Compression via Multiresolution Representation and Predictive Coding", Dept. of Electrical, Computer and System Engineering, Rensselaer Polytechnic Institute, Troy, NY 1993を参照され

40

50

たい。

【 0 0 0 6 】

圧縮は、しばしば非常に時間がかかり、また膨大なメモリを必要とする。より高速に、かつ/又は、可能なかぎり少ないメモリで、圧縮を行うのが望ましい。品質を保証できない、圧縮率が不十分である、あるいはデータレートが制御可能でないという理由で、圧縮を利用しなかった応用分野もある。しかし、伝送及び/又は記憶すべき情報量を減らすため圧縮を利用するのが望ましい。

【 0 0 0 7 】

従来技術に、自然連続階調画像を扱うための圧縮システムがある。その一例が、国際標準 Dis. 10918-1, " Digital Compression and Coding of Continuous-Tone Still Images ", CCITT勧告 T. 8 1 であり、これは通常、J P E G と呼ばれる。従来技術に、2 値/ノイズフリー/浅画素深度画像を扱うための圧縮システムもある。そのようなシステムの一例が、国際標準 I S O / I E C 11544, " Information Technology-Coded Representation of Picture and Audio Information-Progressive Bi-level Image Compression ", CCITT勧告 T. 8 2 であり、これは通常、J B I G と呼ばれる。しかしながら、従来技術には両方を適切に処理するシステムがない。そのようなシステムがあると望ましい。

10

【 0 0 0 8 】

パーサ (parser) はコンピュータ科学において周知である。パーサは、構造が初めは分かっていないオブジェクトの種々の部分に意義付けする役割がある。例えば、コンパイラの一部として動作するあるパーサは、プログラム・ファイル中のある文字列が " 識別子 " であり、別の文字列が予約語を構成し、また別の文字列がコメントの部分であると決定するだろう。このパーサは、文字列がどういう " 意味 " であるかを判定するのではなく、対象のどういう種類の部分であるかを判断するだけである。

20

【 0 0 0 9 】

ほとんどの画像記憶フォーマットは単一用途のものである。すなわち、単一の解像度または単一の品質レベルしか利用できない。他の画像フォーマットは多用途が可能である。従来技術の多用途画像フォーマットの中には、2 つ又は3 つの解像度/品質の選択肢をサポートするものもあるが、解像度又は品質の一方しか指定できず、両方は指定できないものもある。利用できる解像度及び品質の選択肢を増加させることが望ましい。

30

【 0 0 1 0 】

例えば、インターネットのワールド・ワイド・ウェブサーバーは、現在、大量のデータの中から必要とされる情報を提供する。普通、ユーザは画面上で多数の画像を閲覧し、いくつかを印刷することに決められることができる。しかし残念ながら、閲覧ツールの現状では、画像が主にモニタ用のものであると印刷出力の品質がかなり悪くなってしまい、画像が主に印刷用のものであると閲覧時間が極端に長くなってしまふ。 " 非損失の " 画像の取得は、不可能であるか、あるいは、全く独自のダウンロードとしてなされねばならない。

【 0 0 1 1 】

【 発明が解決しようとする課題 】

本発明の一般的な目的は、良好なエネルギー集中をもたらす変換を利用する損失性及び非損失性のデータ圧縮システムを提供することにある。より具体的に述べれば、本発明の目的は、自然連続階調の画像、2 値/ノイズフリー/浅画素深度の画像、及び、その両方の種類のデータを含む画像を適切に処理できるデータ処理システムを提供すること、様々な解像度/品質の画像フォーマットをサポートできるデータ圧縮システムを提供することにある。

40

【 0 0 1 2 】

【 課題を解決するための手段 】

請求項 1 乃至 7 の各項記載の発明によるデータ圧縮システムは、ウェーブレット方式のコード、バイナリ方式のコード、及び、ウェーブレット方式又はバイナリ方式のコードを選択するために接続された選択制御部からなる。そして、請求項 1 記載の発明によれば、

50

ウェーブレット方式のコードは可逆ウェーブレット変換部、該可逆ウェーブレット変換部に接続された埋め込み順序付け量子化器、及び、該埋め込み順序付け量子化器に接続されたコンテキストモデルからなり、また、ウェーブレット方式のコードとバイナリ方式のコードが共有するエントロピー・コードをさらに含む。請求項 2 記載の発明によれば、バイナリ方式はグレイ符号化を実行する。請求項 3 記載の発明によれば、エントロピー・コードは有限状態マシン・コードからなり、請求項 4 記載の発明によれば、有限状態マシン・コードはルックアップテーブルからなる。請求項 5 記載の発明によれば、エントロピー・コードは Q コードからなり、請求項 6 記載の発明によれば、エントロピー・コードは Q M コードからなり、請求項 7 記載の発明によれば並列コードからなる。

#### 【 0 0 1 4 】

請求項 8 記載の発明によるデータ圧縮システムは、ヒストグラム圧縮機構、該ヒストグラム圧縮機構に接続された可逆ウェーブレット変換部、該可逆ウェーブレット変換部に接続された埋め込み順序付け量子化器、該埋め込み順序付け量子化器に接続されたコンテキスト・モデリング機構のウェーブレット方式のコード、バイナリ方式符号化部、及び、該コンテキスト・モデリング機構及びバイナリ方式の符号化部に接続されたエントロピー・コードからなり、さらに、ウェーブレット方式又はバイナリ方式を選択するために接続された選択制御部を含む。請求項 9 記載の発明によればヒストグラム圧縮機構はブーリアンヒストグラムを作り、請求項 10 記載の発明によればヒストグラム圧縮機構は整数値を画像データ中の全ての可能な画素値に写像する。請求項 11 記載の発明によれば、ヒストグラム圧縮機構で利用されたマッピングを復号化器に通知するために接続された通知機構をさらに含む。請求項 12 記載の発明によれば、マッピングは復号化器に受け取られた圧縮データに含まれるヘッダで通知される。請求項 13 記載の発明によれば、ヘッダ中の 1 つのビットが、それがセットされたときに、カレント・タイルのために別のヒストグラムが利用されることを復号化器に指示する。請求項 14 記載の発明によれば、復号化器は、値のダイナミックレンジと等しい、ある数のビットを送ることによって通知され、該ある数のビット中の各ビットはダイナミックレンジ内の対応した値が使われるときにセットされる。

#### 【 0 0 1 6 】

##### 【 発明の実施の形態 】

圧縮及び伸長のための方法及び装置について述べる。以下の本発明に関する詳細な説明において、本発明を完全に理解してもらうために、コードの種類、ビット数、信号名等々、様々な具体例が示される。しかし、当業者には、そのような具体例によらずに本発明を実施し得ることは明白になる。他方、本発明をいたずらに難解にしないため、周知の構造及びデバイスはブロック図の形式で表し、詳しくは示さない。

#### 【 0 0 1 7 】

以下の詳細説明のかなりの部分は、コンピュータメモリ内のデータビットに対する演算のアルゴリズム及び記号表現によって与えられる。このようなアルゴリズム記述及び表現は、データ処理技術分野の当業者によって、その研究の内容を他の当業者に対し最も効率的に伝えるために用いられる手段である。あるアルゴリズムがあり、それが概して、希望する結果に至る自己矛盾のないステップ系列だと考えられるとしよう。これらのステップは、物理量の物理的処理を必要とするものである。必ずという訳ではないが、これらの物理量は記憶、転送、結合、比較、その他処理が可能な電氣的または磁氣的信号の形をとるのが普通である。これらの信号をビット、値、要素、記号、文字、用語、数字等で表わすが、主に慣用上の理由から、時に都合がよいことが分かっている。

#### 【 0 0 1 8 】

しかしながら、このような用語は、適切な物理量と関係付けられるべきであり、また、これら物理量につけた便宜上のラベルに過ぎないということに留意すべきである。以下の説明から明らかなように、特に断わらない限り、“処理” “演算” “計算” “判定” “表示” 等々の用語を用いて論じることは、コンピュータシステムのレジスタ及びメモリ内の物理的（電子的）な量として表現されたデータを処理して、コンピュータシステムのメモリ

10

20

30

40

50

またはレジスタ、同様の情報記憶装置、情報伝送装置あるいは表示装置の内部の同様に物理量として表現された他のデータへ変換する、コンピュータシステムあるいは同様の電子演算装置の作用及びプロセスを指すものである。

【0019】

本発明はまた、本明細書に述べる操作を実行するための装置にも関係する。この装置は、要求目的のために専用で作られてもよいし、あるいは、汎用コンピュータを内蔵プログラムにより選択的に駆動または再構成したものでもよい。本明細書に提示されるアルゴリズム及び表示は、本質的に、いかなる特定のコンピュータやその他装置とも関係がない。様々な汎用マシンを本明細書に述べたところから従うプログラムで利用してもよいし、あるいは、必要な方法ステップの実行のためにより特化した装置を作るほうが好都合であるかもしれない。これら多様なマシンに要求される構造は以下の説明より明らかになる。さらに、本発明を説明するにあたり、いかなる特定のプログラミング言語とも関連付けない。本明細書において述べるように、本発明の教えるところを実現するために多様なプログラミング言語を使用してよいことが分かるであろう。

10

【0020】

下記用語が以下の説明に用いられる。それら各種用語にはすでに語義がある。しかし、規定された語義は、それら用語が当該分野において知られている範囲に限定して考えられるべきでない。これら語義は、本発明の理解を手助けするために規定されたものである。

【0021】

アラインメント (alignment) :

ある周波数帯域内の変換係数の、他の周波数帯域に対するシフト度合。

20

【0022】

バイナリ符号化方式 :

2値、有限画素深度の、又はノイズフリーのデータのための符号化の一方式。一実施例にあっては、バイナリ符号化方式は画素のグレイ (Gray) 符号化と特有のコンテキスト・モデルからなる。

【0023】

ビット・シグニフィカンス (bit-significance) :

符号 (sign) 絶対値表現に似た数表現で、ヘッド (head) ビットの後に符号 (sign) ビットが続き、さらに、テール (tail) ビットがあれば、その後続く。埋め込み (embedding) は、この数表現に対しビットプレーン順に符号化する。

30

【0024】

コンテキスト・モデル :

符号化しようとするカレント・ビットに関する原因として利用可能な情報で、カレント・ビットに関する過去に学習した情報を提供し、エントロピー符号化のための条件付確率予測を可能にする。

【0025】

埋め込み量子化 :

符号ストリームに包含される量子化。例えば、重要性レベルが、最高のレベルから最低のレベルへと順に並べられているときには、符号ストリームの単なる打ち切りによって量子化が行われる。タグ、マーカー、ポインタ、その他の信号によって同じ作用を得ることができる。

40

【0026】

エントロピー・コーダ :

カレント・ビットを、確率予測に基づいて符号化又は復号化する装置。

【0027】

エントロピー・コーダは、本明細書では多重コンテキスト・バイナリ・コーダとも呼ばれるであろう。カレント・ビットのコンテキストは " 近傍 " ビットに関するいくつかの選ばれた配置であり、カレント・ビット (1ビットまたは複数ビット) の最適表現のための確率予測を可能にする。一実施例では、エントロピー・コーダはバイナリ・コーダ又はハフ

50

マン・コーダを含む。

【0028】

固定長：

データの特定ブロックを圧縮データの特定ブロックへ変換する方式。例えばBTC（ブロック打ち切り符号化）、VQ（ベクトル量子化）のいくつかの方式。固定長符号は固定レート・固定サイズのアプリケーションに適するが、レート・歪み性能は可変レート方式に比べ劣ることが多い。

【0029】

固定レート：

ある一定の画素レートを維持しなければならない、帯域幅の限定された通信路を持つアプリケーション又は方式。この目的を成し遂げるには、全体的に平均して圧縮するというよりも、局所的に平均して圧縮することが必要である。例えば、MPEGは固定レートを要求する。

10

【0030】

固定サイズ：

限られたサイズのバッファを持つアプリケーション又は方式。この目的を成し遂げるため、全体的に平均した圧縮が達成される、例えば、印刷バッファ。（アプリケーションは、固定レートでかつ固定サイズのこと、そのどちらかのこともある。）

周波数帯域：

各周波数帯域は、同じフィルタ処理系列によりもたらされる一群の係数を表す。

20

【0031】

ヘッド・ビット：

ビット・シグニフィカンス表現において、ヘッドビットとは、最上位ビットから最初の非ゼロのビットまでの、該最初の非ゼロビットを含めた絶対値ビットである。

【0032】

水平コンテキストモデル：

（一実施例では）埋め込みウエーブレット係数及びバイナリ・エントロピー・コーダのためのコンテキスト・モデル。

【0033】

ベキ等：

画像を損失性形式で伸長してから同じ損失性符号語へ再圧縮することを可能にする符号化。

30

【0034】

画像タイル：

それぞれが同一のパラメータを持つ、オーバーラップのない連続した部分画像の格子の定義を可能にするため選ばれた矩形領域。画像タイルは、ウエーブレット方式符号化において変換の計算のため必要になるバッファ・サイズに影響を及ぼす。画像タイルはランダムにアドレスできる。符号化操作は1画像タイル中の画素及び係数データを処理用する。このため、画像タイルを乱順に構文解析又は復号化することができる。すなわち、画像タイルを、ランダムにアドレスし、又は注目領域の伸長の様々な歪みレベルに応じて復号化

40

【0035】

重要性レベル：

特定の体系を定義することにより、入力データ（画素データ、係数、誤差信号等）は視覚的効果が同じ複数のグループに論理的に分類される。例えば、最上位の一つまたは複数のビットプレーンは、多分、それより下位のビットプレーンより視覚的に重要であろう。また、低い周波数の情報は一般に高い周波数の情報より重要である。“視覚的重要性”の実用定義の殆どは、後述のように本発明も含め、何らかの誤差基準に関係している。しかし、それよりも良好な視覚的尺度が、視覚的重要性の体系定義に組み入れられるかもしれな

50

い。データの種類が異なれば視覚的重要性レベルも異なる。例えば、音声データは音声の重要性レベルを持つ。

【0036】

オーバーラップ変換：

単一のソース標本点が同一周波数の複数の係数に寄与する変換。その例に、多くのウエーブレットとオーバーラップ直交変換 (Lapped Orthogonal Transform) がある。

【0037】

プログレッシブ：

符号化データの一部から矛盾のない伸長結果を得られ、かつデータを増やすことで精度を上げることができるように順序付けられた符号ストリーム。データのビットプレーンが浅いほうから深いほうへ順序付けられた符号ストリーム；この場合は、普通、ウエーブレット係数データをさす。

10

【0038】

プログレッシブ画素深度：

データのビットプレーンが浅いほうから深いほうへ順序付けられた符号ストリーム。

【0039】

プログレッシブ・ピラミッド：

解像度が下がる毎に大きさが2分の1 (面積では4分の1) になる解像度成分の連続。

【0040】

可逆変換：

20

一実施例では、圧縮結果を元に復元できる、整数演算により実施される効率的変換。

【0041】

S変換：

1つの2タップ・ローパスフィルタと1つの2タップ・ハイパスフィルタからなる特殊な可逆ウエーブレットフィルタ対。

【0042】

テール：

ビット・シグニフィカンス表現で、テール(tail)ビットとは最上位の非ゼロのビットより下位の低い絶対値ビットである。

【0043】

30

テール情報：

一実施例では、ビット・シグニフィカンス表現で表された係数のためにとり得る4つの状態。係数及びカレント・ビットプレーンの関数であり、水平コンテキスト・モデルのために利用される。

【0044】

テール・オン(tail-on)：

一実施例では、テール情報の状態がゼロか非ゼロであるかに依存した2つの状態。水平コンテキスト・モデルのために利用される。

【0045】

タイルデータ(tile data)セグメント：

40

一つの画像タイルを完全に記述する符号ストリームの部分。一実施例においては、画像タイルの始まり(SOT)を定義するタグから、次のSOTまで、又は画像の終わり(EOI)のタグまでの全データ。

【0046】

変換係数：

ウエーブレット変換を適用した結果。ウエーブレット変換においては、係数は対数分割された周波数スケールを表す。

【0047】

TS変換：

2・6 (Two-Six) 変換。1つの2タップ・ローパス分析フィルタと1つの6タップ・ハ

50



イパス分析フィルタからなる特殊な可逆ウエーブレットフィルタ対。合成フィルタは、分析フィルタの直交ミラー・フィルタである。

【 0 0 4 8 】

TT変換：

2・10 (Two-Ten) 変換。1つの2タップ・ローパス分析フィルタと1つの10タップ・ハイパス分析フィルタからなる特殊な可逆ウエーブレットフィルタ対。合成フィルタは分析フィルタの直交ミラー・フィルタである。

【 0 0 4 9 】

統合型 (unified) 非損失性 / 損失性：

同じ圧縮システムが、非損失性又は損失性の復元が可能な符号データストリームを提供する。

10

【 0 0 5 0 】

ウエーブレット・フィルタ：

ウエーブレット変換に使われるハイパスとローパスの合成フィルタ及び分析フィルタ。

【 0 0 5 1 】

ウエーブレット変換：

”周波数”及び”時間(空間)”領域の両方の拘束条件を用いる変換。説明する一実施例では、1つのハイパスフィルタと1つのローパスフィルタからなる変換である。結果として得られる係数は2:1の間引きを施され(臨界フィルタ処理)、次にそれらフィルタがローパス係数にかけられる。

20

【 0 0 5 2 】

ウエーブレット・ツリー：

最高レベルのウエーブレット分解のLL部内の単一の係数と関係付けられた係数群。係数の個数はレベル数の関数である。ウエーブレット・ツリーのスパンは、分解レベル数に依存する。例えば、1レベル分解の場合には、ウエーブレット・ツリーのスパンは4画素、2レベル分解では16画素、等々である。

【 0 0 5 3 】

本発明の概要

本発明は、符号化部及び復号化部を持つ圧縮/伸長システムを提供する。符号化部は入力データを符号化して圧縮データを生成する働きをし、他方、復号化部は既に符号化されたデータを復号化して元の入力データの再構成データを生成する働きをする。入力データには、画像(静止画像あるいは動画像)、音声等々の様々な種類のデータが含まれる。一実施例では、データはデジタル信号データであるが、デジタル化したアナログデータ、テキストデータ形式、その他の形式も可能である。そのデータのソースは、例えば符号化部及び/または復号化部のためのメモリまたは通信路である。

30

【 0 0 5 4 】

本発明において、符号化部及び/または復号化部の構成要素は、ハードウェア又はコンピュータシステム上で利用されるソフトウェアによって実現し得る。本発明は、非損失性の圧縮/伸長システムを提供する。本発明はまた、損失性の圧縮/伸長を実行するようにも構成し得る。本発明は、圧縮データの構文解析を、伸長をすることなく実行するように構成し得る。

40

【 0 0 5 5 】

本発明のシステムの概要

本発明は、自然画像に見られる滑らかなエッジと平坦な領域を非常に良好に表現する。本発明は、可逆埋め込みウエーブレットを利用して、画素深度の深い画像を圧縮する。しかしながら、可逆埋め込みウエーブレット、他のウエーブレット変換方式及びシヌソイド変換方式は、テキストや図形画像に見られるシャープなエッジを表現するのは得意ではない。この種の画像は、グレイ(Gray)符号化を行ってからJBIGのようなコンテキストベースのビットプレーン符号化を行うことにより良好に圧縮できる。さらに、ノイズフリーのコンピュータ生成画像は、バイナリ方式により良好にモデル化される。

50

## 【 0 0 5 6 】

本発明は、2値画像及び図形画像の圧縮のためのバイナリ方式を提供する。このバイナリ方式は、ダイナミックレンジ全体を使わないある種の画像に対する圧縮も改善する。このバイナリ方式においては、本発明は変換を使わないで画像のビットプレーンを符号化する。

## 【 0 0 5 7 】

図1は、バイナリ方式を採用した本発明の圧縮システムの一実施例のブロック図である。なお、システムの復号化部は逆の順序で動作し、データフローも同様である。図1において、入力画像101は多成分処理機構111に入力される。この多成分処理機構111は、オプションの色空間変換、及び、サブサンプリングを施された画像成分に関するオプションの処理を提供する。方式選択機構110は、画像が連続階調画像か2値画像か、あるいは、画像のどの部分がそのような特性を持っているかを判定する。画像データは方式選択機構110へ送られ、方式選択機構110は、その画像データ又はその部分をウェーブレット方式処理(ブロック102, 103, 105)又はバイナリ方式処理(ブロック104)へ送る。本発明においては、どのモードを利用するかは、データに依存して決まる。一実施例では、方式選択機構110はマルチプレクサからなる。方式選択機構110は、復号化動作中は利用されない。

10

## 【 0 0 5 8 】

ウェーブレット方式では、可逆ウェーブレット変換ブロック102が可逆ウェーブレット変換を実行する。同ブロック102の出力は係数の系列である。埋め込み順序付け量子化ブロック103は、(可逆ウェーブレット変換ブロック102により生成された)入力画像101中の係数全部のアラインメントを生成するため、係数をビット・シグニフィカンス表現にしてからラベル付けする。

20

## 【 0 0 5 9 】

画像データ101が受け取られ、そして(適切な多成分処理の後)可逆ウェーブレット変換ブロック102において後に説明されるように可逆ウェーブレットを利用して変換されることにより、画像の多重解像度分解を表す係数の系列が生成される。本発明の可逆ウェーブレット変換は、計算が複雑でない。この変換は、ソフトウェア又はハードウェアにより、全く系統誤差を生じさせないで実行できる。さらに、本発明のウェーブレットはエネルギー集中及び圧縮性能に優れている。これらの係数は埋め込み順序付け量子化ブロック103に受け取られる。

30

## 【 0 0 6 0 】

埋め込み順序付け量子化ブロック103は、後述のように埋め込み順序付け量子化をする。その結果は埋め込み(embedded)データストリームである。この埋め込みデータストリームは、符号化時、伝送時又は復号化時に、結果の符号ストリームの量子化を許す。一実施例においては、埋め込み順序付け量子化ブロック103は、係数を順序付けして符号・絶対値形式に変換する。

## 【 0 0 6 1 】

埋め込みデータストリームは水平コンテキストモデル・ブロック105に受け取られる。水平コンテキストモデル・ブロック105は、埋め込みデータストリーム中のデータをその重要性に基づきモデル化する(後述)。変換モードでは、"ビットプレーン"は変換係数の重要性レベル・プレーンであり、水平コンテキストモデル・ブロック105はウェーブレット係数をビット・シグニフィカンス表現に整える。

40

## 【 0 0 6 2 】

順序付け及びモデリングの結果は、エントロピー・コード106により符号化すべきデシジョン(decisions)(又はシンボル)である。一実施例では、全てのデシジョンが一つのコードへ送られる。他の実施例では、デシジョンは重要性によってラベル付けされ、各重要性レベルのデシジョンは別々の複数の(物理または仮想)コードによって処理される。ビットストリームは、エントロピー・コード106により重要性の順に符号化される。一実施例では、エントロピー・コード106は1つ又は複数のバイナリ・エントロピー・コ

50

ーダからなる。別の実施例では、ハフマン符号化が利用される。

【0063】

バイナリ方式では、グレイ(Gray)符号化ブロック104が入力画像101の画素に対しグレイ符号化を行う。グレイ符号化は画素のビットプレーン間の相関の一部を利用するビット操作である。それは、任意の値 $x$ と $x+1$ に対し、

$\text{gray}(x)$ と $\text{gray}(x+1)$ は<その基数2の表現で異なるのは1ビットだけであるからである。

一実施例では、グレイ符号化ブロック104は8ビット画素に対し点毎の変換、すなわち $\text{gray}(x) = x \text{ XOR } x/2$

を実行する。本発明は、この形式のグレイ符号化を利用することに限定されるわけでも、8ビットのサイズの画素を利用しなければならないわけでもない。しかし、上記式を利用すると、ビットプレーン単位のプログレッシブ伝送の場合のように、利用可能な最上位ビットの一部だけで画素を再構成できるという利点がある。言い換えると、この形式のグレイ符号化はビット・シグニフィカンスの順序付けを保存する。

10

【0064】

バイナリ方式では、グレイ符号化ブロック104及びエントロピー・コード106を利用し、データはビットプレーン毎に符号化される。一実施例では、グレイ符号化ブロック104内のコンテキストモデルは、カレント・ビットを、空間及び重要性レベル情報を利用して条件付けする。

【0065】

バイナリ方式の場合、グレイ符号化画素に対しJBIGのようなコンテキストモデルが利用される。一実施例においては、画像タイルの各ビットプレーンは別々に符号化され、それぞれのビットは、周辺の10画素の値を利用し、ラスタ順に条件付けされて符号化される。図2はバイナリ方式における各ビットプレーンの各ビットのためのコンテキストモデルの幾何学的関係を示す。このビットの条件付けは、固有パターン毎の適応的確率予測をもたらす。なお、バイナリ・エントロピー・コードがグレイ符号化値のビットプレーン・エントロピー符号化に利用されるときには、いくつかのテンプレートがバイナリ・エントロピー・コードのコンテキストモデルのために用いられてもよい。図3は $2^9$ 個のコンテキスト・ビン(bin)のための7画素と2ビットのビットプレーン情報を示す。

20

【0066】

このコンテキストとカレント・ビットの値を利用して、エントロピー・コード106はビットストリームを生成する。この同じバイナリ・エントロピー・コード106が、変換モードとバイナリ方式の両方のデータの符号化に利用される。一実施例では、エントロピー・コード106はルックアップ・テーブルで実現される有限状態マシンからなる。なお、本発明は、Qコード、QMコード、高速並列コードのような任意のバイナリ・エントロピー・コードと一緒に利用し得る。

30

【0067】

エントロピー・コード106はいずれの方式についても同じものであり、かつ、バイナリ・コンテキストモデルは単純であるため、同一システムでバイナリ方式と変換方式を実現するのに、ごく僅かな追加資源しか必要とされない。さらに、コンテキストモデルの構成は異なるが、両モードのための必要資源は同じである。すなわち、両方のモードで、コンテキスト格納のために同じメモリを利用し、また、同じバイナリ・エントロピー・コードを利用する。

40

【0068】

本発明は、画像全体に対して実行されてもよいし、あるいは、より一般的であるが、画像のタイリングされたセグメントに対して実行されてもよい。タイルの中には、変換方式による方が圧縮が良好なものと、バイナリ方式による方が良好なものとがある。使用すべきモードの選択アルゴリズムは様々なものが可能である。タイルが利用されるときには、タイル単位のランダムアクセスが可能である。また、注目領域は各別に高精度に復号化することができる。最後に、変換方式とバイナリ方式のいずれを選択するかは、1つ1つのタイル毎に決定することができる。

50

## 【 0 0 6 9 】

また、画像は、本発明のデュアルモード・システムを利用してビットプレーンに関しプログレッシブであり、J B I Gに教えられるように階層形式に符号化し得ることに注意されたい。

## 【 0 0 7 0 】

復号化に関しては、タイルのヘッダ中の1ビットを、データの符号化に利用された方式を指定するために利用してよい。方式選択機構110は用いられない。元のダイナミックレンジから低いダイナミックレンジへの非損失性マッピング、例えばヒストグラム圧縮(後述)によるようなものが可能であれば、さらに役に立つことがある。J B I Gにおけるようなルック・アヘッド(look ahead)を利用してよい。このルック・アヘッドは、J B I Gにおけるような普通の予測又は決定論的予測を使用してよい。

10

## 【 0 0 7 1 】

バイナリ方式又は変換方式の選択

方式選択機構110は、バイナリ方式と変換方式の選択をする。一実施例では、入力画像は両方の方式で符号化され、方式選択機構110は得られたビットレートが低い方の方式を選択する(非損失性圧縮を仮定)。つまり、よく圧縮する方のモードが選ばれる。この方法は、コストが高いと思われるかもしれないが、それほどではない。というのは、バイナリ方式と変換方式は共に、ソフトウェアが比較的高速であり、かつ、ハードウェアも小規模であるからである。この方法から派生する方法は、コードをバイパスし、エントロピー値を利用して低い方のビットレートを判定する方法である。

20

## 【 0 0 7 2 】

別の実施例においては、本発明は画像の画素値の完全な(又は部分的な)ヒストグラム、又は、隣接画素値のペア間の差分のヒストグラムを生成する。画素値のヒストグラムの場合、そのヒストグラムのピークが、画素深度のダイナミックレンジより遥かに小さな値で生じたならば、バイナリ方式を選ぶ、というようなデータの統計的解析を使う。

## 【 0 0 7 3 】

一実施例では、本発明は隣接画素のペア間の第1次差分の完全な(又は部分的な)ヒストグラムが生成される。標準的な画像では、そのようなヒストグラムは正にラプラシアン分布であり、ウェーブレット方式が利用されよう。しかし、ヒストグラムがラプラシアン分布のピークを持たないときには、バイナリ方式が利用される。

30

## 【 0 0 7 4 】

両方の種類のヒストグラムを生成し、方式の選択のために一緒に利用してもよい。

## 【 0 0 7 5 】

いずれも後述するが、TS変換又はTT変換のdnフィルタ出力は第1次統計量に近い。これは、変換が実行されヒストグラムが生成される方法を示唆する。そのヒストグラムに基づいて、方式が選択される。その方式が変換方式のときには、システムは既に生成された変換係数を続けて処理する。バイナリ方式が選択されると、変換係数は捨てられ(又は、画素がセーブされたか否かによっては逆変換され)、システムはバイナリ方式を開始する。

## 【 0 0 7 6 】

別の実施例では、領域分割及び/又は文書種類に関する以前の知識が、どちらの方式を選択すべきの決定を支援するかもしれない。

40

## 【 0 0 7 7 】

もっとより長い符号化時間を利用できるならば、2方式の利点を最大にするようにタイリングのサイズを選ぶことができる。

## 【 0 0 7 8 】

なお、一実施例においては、本発明のシステムはバイナリ方式符号化を含まず、したがって、可逆埋め込みウェーブレット圧縮(CREW)及び伸長だけを利用することに注意されたい。

## 【 0 0 7 9 】

50

### ウェーブレット分解

本発明は、最初に、可逆ウェーブレットを利用して、（画像データとしての）画像または他のデータ信号の分解を実行する。本発明において、可逆ウェーブレット変換は、整数係数を持つ信号の非損失性復元が可能な完全再構成システムを整数演算で実現する。効率的な可逆変換は、行列式が1（又はほぼ1）の変換行列によるものである。

#### 【0080】

本発明は、可逆ウェーブレットを利用することにより、有限精度の演算で非損失性圧縮を提供することができる。画像データに可逆ウェーブレット変換を適用することにより生成される結果は、係数の系列である。

#### 【0081】

本発明の可逆ウェーブレット変換は、フィルタの集合を用いて実現し得る。一実施例では、そのフィルタは1つの2タップ・ローパスフィルタと1つの6タップ・ハイパスフィルタである。一実施例では、これらフィルタは加減算（とハードワイヤのビットシフト）だけで実現される。

#### 【0082】

Hadamard変換を利用する本発明の一実施例は、完全再構成システムである。

#### 【0083】

Hadamard変換に関する情報を得るには、Anil K. Jain, "Fundamentals of Image Processing", P.155を読みたい。Hadamard変換の逆変換は、本明細書においてS変換と呼ばれる。

#### 【0084】

S変換は、一般添数nを用いて出力を次のように定義することができる。

#### 【0085】

##### 【数1】

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = x(2n) - x(2n+1) \end{cases}$$

#### 【0086】

なお、変換係数アドレッシングにおける因数2は、暗黙の1/2サブサンプリングの結果である。この変換は可逆であり、その逆変換は次の通りである。

#### 【0087】

##### 【数2】

$$\begin{cases} x(2n) = s(n) + \left\lfloor \frac{d(n)+1}{2} \right\rfloor \\ x(2n+1) = s(n) - \left\lfloor \frac{d(n)}{2} \right\rfloor \end{cases}$$

#### 【0088】

記号

#### 【0089】

##### 【外1】

10

20

30

40

[.]

【0090】

は、切り捨てて丸めること、つまり打ち切りを意味し、床関数と呼ばれることがある。同様に、天井関数

【0091】

【外2】

[.]

10

【0092】

は最も近い整数へ切り上げて丸めることを意味する。

【0093】

完全再構成システムのもう一つの例は2・6(TS)変換である。可逆TS変換は、ローパスとハイパスのフィルタの2つの出力に関する次の式により定義される。

【0094】

【数3】

20

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = \left\lfloor \frac{\left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor + 4(x(2n+2) - x(2n+3)) + \left\lfloor \frac{x(2n+4) + x(2n+5)}{2} \right\rfloor}{4} \right\rfloor \end{cases}$$

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = x(2n+2) - (2n+3) + \left\lfloor \frac{-s(n) + s(n+2) + 2}{4} \right\rfloor \end{cases}$$

30

【0095】

TS変換は可逆であり、その逆変換は次の通りである。

【0096】

【数4】

$$\begin{cases} x(2n) = s(n) + \left\lfloor \frac{p(n)+1}{2} \right\rfloor \\ x(2n+1) = s(n) - \left\lfloor \frac{p(n)}{2} \right\rfloor \end{cases}$$

40

【0097】

ここで、次式により  $p(n)$  がまず計算されなければならない。

【0098】

【数5】

$$p(n) = d(n-1) - \left[ \frac{-s(n-1) + s(n+1) + 2}{4} \right]$$

【 0 0 9 9 】

ローパスフィルタからの結果を、ハイパスフィルタにおいて2度（第1項と第2項で）利用できる。したがって、ほかに2つの加算を行うだけで、ハイパスフィルタの結果を得られる。

【 0 1 0 0 】

完全再構成システムのもう一つの例は2・10（TT）変換である。可逆TT変換は、ローパスとハイパスのフィルタの2つの出力に関する次式により定義される。 10

【 0 1 0 1 】

【 数 6 】

$$d(n) = \frac{\left[ \begin{array}{l} 3 \left[ \frac{x(2n) + x(2n+1)}{2} \right] - 22 \left[ \frac{x(2n+2) + x(2n+3)}{2} \right] \\ + 64(x(2n+4) - x(2n+5)) + 22 \left[ \frac{x(2n+6) + x(2n+7)}{2} \right] \\ - 3 \left[ \frac{x(2n+8) + x(2n+9)}{2} \right] \end{array} \right]}{64} \quad 20$$

【 0 1 0 2 】

このd(n)の式はs(n)を使って単純化することができる（さらに、64による整数除算は、分子に32を足すことにより丸めることができる）。これにより次式が得られる。

【 0 1 0 3 】

【 数 7 】

$$\left\{ \begin{array}{l} s(n) = \left[ \frac{x(2n) + x(2n+1)}{2} \right] \\ d(n) = x(2n+2) - (2n+3) + \left[ \frac{3s(n) - 22s(n+2) + 22(5n+4) - 3s(n+6) + 32}{64} \right] \end{array} \right.$$

【 0 1 0 4 】

このTT変換は可逆であり、その逆変換は次式である。 40

【 0 1 0 5 】

【 数 8 】

$$\left\{ \begin{array}{l} x(2n) = s(n) + \left[ \frac{p(n)+1}{2} \right] \\ x(2n+1) = s(n) - \left[ \frac{p(n)}{2} \right] \end{array} \right.$$

【 0 1 0 6 】

ここで、 $p(n)$  は次式によりまず計算されなければならない。

【 0 1 0 7 】

【 数 9 】

$$p(n) = d(n-1) - \left[ \frac{3S(n) - 22S(n+2) + 22(5n+4) - 3S(n+6) + 32}{64} \right]$$

10

【 0 1 0 8 】

T S 変換と T T 変換のいずれにおいても、S 変換と同様、ローパスフィルタは、入力信号  $x(n)$  のレンジが出力信号  $s(n)$  のレンジと同じになるように作られる。すなわち、平滑出力の増大はまったくない。入力信号が  $b$  ビットの深さのときには、出力信号も  $b$  ビットの深さである。例えば、信号が 8 ビット画像の場合、ローパスフィルタの出力も 8 ビットである。このことは、例えばローパスフィルタを連続して適用することにより平滑出力がさらに分解されるピラミッド・システムのために重要な特性である。従来技術のシステムにおいては、出力信号のレンジが入力信号のレンジより大きく、このことがフィルタの連続的適用を困難にしている。また、変換を整数演算で行う際の丸めによる系統誤差がないので、損失性システムの全ての誤差を量子化により制御可能である。さらに、ローパスフィルタは、2 つのタップしか持たないため、非オーバーラップ・フィルタになる。この特性は、ハードウェア化のために重要である。

20

【 0 1 0 9 】

一実施例では、3 と 2 2 による乗算は、図 2 2 に示すようなシフトと加算により実現される。図 2 2 において、 $s(n)$  入力は乗算器 1 5 0 1 に接続され、乗算器 1 5 0 1 は  $s(n)$  入りに 2 を乗じる。一実施例では、この乗算は、 $s(n)$  信号のビットの 1 桁左シフトとして実現される。乗算器 1 5 0 1 の出力は加算器 1 5 0 2 により  $s(n)$  信号と加算される。加算器 1 5 0 2 の出力は、3  $s(n)$  信号である。加算器 1 5 0 2 の出力はまた、乗算器 1 5 0 3 により 2 を乗じられる。乗算器 1 5 0 3 は、1 桁左シフトとして実現される。乗算器 1 5 0 3 の出力は加算器 1 5 0 5 により乗算器 1 5 0 4 の出力と加算され、乗算器 1 5 0 4 は  $s(n)$  信号を 4 桁左シフトにより 1 6 倍する。加算器 1 5 0 5 の出力は 2 2  $s(n)$  信号である。

30

【 0 1 1 0 】

フィルタに対する厳格な可逆性要件は、次のことに着目することによって緩和することができる。ハイパス係数は、ある順序で符号化されて復号化される。前に復号化されたハイパス係数に対応する画素値は、正確に分かっているので、カレント・ハイパスフィルタ処理に用いることができる。

【 0 1 1 1 】

T S 変換及び T T 変換は非オーバーラップのローパス合成フィルタとハイパス分析フィルタを有する。ハイパス合成フィルタとローパス分析フィルタだけがオーバーラップ・フィルタである。

40

【 0 1 1 2 】

T S フィルタは、タイル境界に関し良好な特性を有する。タイルのサイズがツリーのサイズの倍数である場合を考える。そして、画像がタイルに分割される場合に生じるような、ある信号の一部に対する変換の適用を考察する。ローパス分析フィルタはオーバーラップ・フィルタではないので、ローパス係数はタイリングによる影響を受けない。すなわち、信号のその部分が均一個数の信号を持つ場合、そのローパス係数は信号全体を変換したときのそれと同じである。

【 0 1 1 3 】

復号化中に、量子化のためハイパス係数が存在せず、画像が S S 係数だけを使って最高圧

50



縮率で再構成される場合には、タイル境界にまたがってローパス合成フィルタが使われてよく、逆変換はローパス係数を用いて信号全体について実行される。SS係数のタイリングによって変化しないので、解はタイリングが利用されないときと全く同じである。これにより、信号の一部に対しフォワード変換を実行することにより生じるアーティファクト (artifacts) が除去される。

【0114】

復号化中に、ハイパス係数が存在する（しかし、ハイパス係数はそれらの値が幾分の不確定性を持つように量子化されている）場合には、オーバーラップ1Dローパス分析フィルタ演算がある境界を横切って別の部分へ入る場所の標本に対し次のことを行うことができる。タイル境界を横切らない標本について、実際に利用されたフィルタに基づき、可能な最小及び最大の再構成値が決定される。その標本に関し、ローパス係数（とローパスフィルタ）だけを用い、そしてタイル境界を横切ることによって、そうであったであろう再構成値（すなわちオーバーラップ予測値）が決定される。そのオーバーラップ予測値が可能な最小と最大の再構成値の間（それらの値も含む）であれば、そのオーバーラップ予測値が用いられる。そうでなければ、その可能な最大と最小の再構成値のうちの、オーバーラップ予測値に近い方の値が用いられる。こうすることによって、信号の断片に対しフォワード変換を実行することにより生じるアーティファクトを減らす。

10

【0115】

1Dフィルタ演算が実行される度に再構成値が選ばれる。これが正しく行われると、各ハイパス係数はちょうど一つの有効な再構成値が与えられ、選択により誤差を変換の複数レベルに伝搬し得なくなる。

20

【0116】

非線形画像モデル

本発明の一実施例は、TS変換又はTT変換のような線形フィルタの可逆近似であるウェーブレット・フィルタを利用する。一実施例では、可逆非線形フィルタが用いられるかもしれない。TS変換及びTT変換に類似した非線形フィルタの一種は次のとおりである。

【0117】

【数10】

$$s(n) = \lfloor (x(2n) + x(2n+1)) / 2 \rfloor$$

30

$$d(n) = x(2n) - x(2n+1) + q(n)$$

【0118】

その逆変換はTS変換及びTT変換の場合と同じであるが、ただしp(n)は次の通りである。

【0119】

【数11】

$$p(n) = d(n-1) - q(n)$$

40

【0120】

この実施例において、q(n)は平滑(smooth)係数（及び必要なら前の詳細(detail)係数）からのx(2n)-x(2n+1)に関する予測である。この予測は非線形画像モデルを利用する。一実施例では、非線形画像モデルはHuber-Markov確率場である。この非線形画像モデルは、フォワード変換と逆変換sにおいて全く同一である。反復画像モデルの場合、反復の回数と次数は同一である。

【0121】

非線形画像モデルの一例は次の通りである。ある一定の反復回数の場合に、それぞれの値（画素又はローパス係数）x(k)は新たな値x'(k)に調整される（ただし、kは2n又は

50

2n + 1である)。任意の反復回数を用いてよいが、一実施例では3回の繰り返しが用いられる。最終の反復から q(n)の値が求まる。

【0122】

【数12】

$$q(n) = x'(2n) - x'(2n+1)$$

【0123】

反復の度に、各 x(k)に対する変化 y(k)が計算される。

【0124】

【数13】

10

$$y(k) = -A * \sum_i H(B_i(k))$$

【0125】

ここでAは変化率であり、これは任意の正值でよい。一実施例ではA = 1である。B<sub>i</sub>は差分検出子である。例えば、1次元の場合、B<sub>i</sub>は次の通りである。

【0126】

【数14】

20

$$B_i(k) = -x(k-1)/2 + x(k) - x(k+1)/2$$

【0127】

2次元の場合、水平、垂直、及び2つの対角線方向の差分を検出するための4つのB<sub>i</sub>値があろう。B<sub>i</sub>として他の差分演算子を用いてもよい。

【0128】

【数15】

30

$$H(B_i(k)) = \begin{cases} B_i(k) & \text{if } |B_i(k)| < T \\ T & \text{otherwise} \end{cases}$$

【0129】

ここで、Tは閾値であり、これは任意の正值でよい。Tは画像のエッジを構成する差分を表す。一実施例では、T = 8である。

【0130】

$x'(2n) + x'(2n+1) = x(2n) + x(2n+1)$ の拘束条件のもとに、値  $x'(2n)$ 、 $x'(2n+1)$  のペアは変化  $y(2n)$ 、 $y(2n+1)$  のペアによって調整される。これは、変化  $y(2n)$ 、 $y(2n+1)$  を結合して単一の変化  $y'(2n)$ 、つまり、両変化にサポートされる最大の変化にすることによって達成される。

40

【0131】

y(n)とy(2n+1)が両方とも正か負のときには、

$$y'(2n) = 0$$

である。そうでない場合、 $|y(2n)| < |y(2n+1)|$  ならば、

$$y'(2n) = y(2n)$$

であり、そうでなければ

$$y'(2n) = -y(2n+1)$$

である。そして

50

$$x'(2n) = x(2n) + y'(2n)$$

$$x'(2n+1) = x(2n+1) - y'(2n)$$

である。Huber - Markov確率場に関するこれ以上の情報を得るには、

R . R . Schultz and R . L . Stevenson , " Improved definition image expansion " , Proceedings of IEEE International Conference on Acoust. , Speech and Signal Processing , vol . III , pp.173-176 , San Francisco , March 1992 を見られたい。

#### 【 0 1 3 2 】

いくつかの実施例では、変換は1次元から2次元へ拡張されるが、その拡張のためにまず  $q(n) = 0$  として各次元に対し別々に変換を行う。次に、画像モデルの同様の適用によっ

10

#### 【 0 1 3 3 】

##### 2次元ウェーブレット分解

本発明のローパスフィルタ及びハイパスフィルタを用いて、多重解像度分解が行なわれる。分解レベル数は可変であり任意数でよいが、現在のところ分解レベル数は2レベル乃至5レベルである。最大レベル数は、 $\log_2$ (長さ又は幅の最大値)である。

#### 【 0 1 3 4 】

画像のような2次元データに対し変換を実行する最も普通のやり方は、1次元フィルタを別々に適用する方法、つまり、行に沿って適用したのち列に沿って適用するという方法である。第1レベルの分解により4つの異なった係数バンドが得られ、これら係数バンドは

20

#### 【 0 1 3 5 】

ウェーブレット分解の各周波数サブバンドはさらに分解することができる。最も普通のやりかたはLLサブブロックだけをさらに分解する方法であり、これは各分解レベルのLL周波数サブバンドが生成された時にそれをさらに分解することを含むであろう。このような多重分解はピラミッド分解と呼ばれる(図4乃至図7)。記号LL, LH, HL, HH

30

#### 【 0 1 3 6 】

例えば、可逆ウェーブレット変換が再帰的に1つの画像に適用されると、第1レベルの分解は最も細かいディテールもしくは解像度に対し作用する。第1分解レベルで、画像は4つのサブ画像(すなわちサブバンド)に分解される。各サブバンドは、1つの空間周波数帯域を表わしている。第1レベルのサブバンドはLL0, LH0, HL0, HH0と表される。元の画像を分解するプロセスは、水平、垂直の両次元における1/2サブサンプリングを含むので、図4に示されるように、第1レベルのサブバンドLL0, LH0, HL0, HH0はそれぞれ、入力を持っていた画像の画素(または係数)の個数の4分の1の個数の

40

#### 【 0 1 3 7 】

サブバンドLL0は、水平方向の低い周波数情報と垂直方向の低い周波数情報を同時に含んでいる。一般に、画像エネルギーの大部分は当該サブバンドに集中している。サブバンドLH0は、水平方向の低い周波数情報と垂直方向の高い周波数情報(例えば水平方向エッジの情報)を含んでいる。サブバンドHL0は、水平方向の高い周波数情報と垂直方向の低い周波数情報(例えば垂直方向エッジの情報)を含んでいる。サブバンドHH0は、水平方向の高い周波数情報と垂直方向の高い周波数情報(例えばテクスチャ又は斜めエッジの情報)を含んでいる。

#### 【 0 1 3 8 】

50

この後に続く第2、第3、第4の下位分解レベルはそれぞれ、前レベルの低周波数LLサブバンドを分解することによって作られる。第1レベルの当該サブバンドLL0が分解されることによって、図5に示すように、やや精細な第2レベルのサブバンドLL1, LH1, HL1, HH1が作られる。同様に、サブバンドLL1が分割されることによって、図6に示すように、精細度の粗い第3レベルのサブバンドLL2, LH2, HL2, HH2が生成される。また、図7に示すように、サブバンドLL2が分割されることにより、精細度がより粗い第4レベルのサブバンドLL3, LH3, HL3, HH3が作られる。2:1のサブサンプリングにより、第2レベルの各サブバンドは、原画像の1/6の大きさである。このレベルの各標本(つまり画素)は、原画像中の同一位置のやや細かいディテールを表す。同様に、第3レベルの各サブバンドは、原画像の1/4の大きさである。第3レベルでの各画素は、原画像中の同一位置のかなり粗いディテールを表す。また、第4レベルの各サブバンドは原画像の1/16の大きさである。

10

## 【0139】

分解画像はサブサンプリングのため原画像より物理的に小さいので、原画像の格納のために使用されたメモリを利用して、分解サブバンド全部を格納できる。つまり、3レベル分割の場合、原画像と分解サブバンドLL0, LL1は捨てられ、保存されない。

## 【0140】

4つのサブバンド分解レベルだけを示したが、個々のシステムの要件に応じて、それ以上のレベルを生成することも可能である。また、DCTのような他の変換又は次元配置のサブバンドによって、様々な親子関係を定義してもよい。

20

## 【0141】

## ピラミッド分解

ウェーブレット分解の各周波数サブバンドはさらに分解することができる。一実施例においては、LL周波数サブバンドだけが分解される。本明細書では、このような分解はピラミッド分解と呼ばれる。記号LL, LH, HL, HHと分解レベル番号で各分解を表示する。なお、本発明のウェーブレット・フィルタによれば、ピラミッド分解は係数サイズを増大させない。

## 【0142】

別の実施例では、LLに加えて他のサブバンドも分解されるかもしれない。以下の説明において、“LL”なる用語は“SS”(“L”=“S”)と入れ替えて用いてもよい。同様に、“H”なる用語も“D”と入れ替えて用いてもよい。

30

## 【0143】

## ウェーブレットのツリー構造

ピラミッド分解のウェーブレット係数には自然で有用なツリー構造がある。なお、LL周波数サブブロックは最終の分解レベルに対応したただ一つしかない。これに対し、LH, HL, HHのバンドはレベル数と同数存在する。このツリー構造により、ある周波数帯域内の係数の親は、それより低い解像度の同じ周波数帯域内の係数であり、かつ同じ空間位置関係にあることが明らかになる。

## 【0144】

各ツリーのルートは、純粋に平滑な係数である。画像のような2次元の信号の場合、ツリーのルートは3つの“子”を持ち、ほかのノードはそれぞれ4つの子を持つ。この階層的ツリーは2次元信号に限定されない。例えば、1次元信号の場合、ルートは1つの子を持ち、ルート以外のノードはそれぞれ2つの子を持つ。これ以上高い次元は、1次元の場合及び2次元の場合より導かれる。

40

## 【0145】

図8は連続した2レベル間の親子関係を表している。図8において、Aの係数は、B, C, Dに対する直接の親であるが、B, C, Dを親とする係数(EとH, FとI, GとJ)に対する親でもある。例えば、Bは、E付近の4係数、H付近の16係数、等々に対する親である。

## 【0146】

50

多重解像度分解のプロセスは、フィルタ系列を使って遂行し得る。

【0147】

1次元の模範的フィルタを使って実現される1次元2レベル変換の例については、米国特許出願第08/498,695号(1995年6月30日受理、“Method and Apparatus For Compression Using Reversible Wavelet Transforms and an Embedded Costream”)及び米国特許出願第08/498,036号(1995年6月30日受理“Reversible Wavelet Transform and Embedded Code-stream Manipulation”)を見られたい。

【0148】

フォワード・ウエーブレット変換の実行

本発明では、水平方向の1-D(次元)パス、次で垂直方向の1-Dパスによってウエーブレット変換が実行される。レベル数で反復回数が決まる。図9は前に定義したようなフォワードTT変換フィルタを使う4レベル分解を表す。

【0149】

別の実施例では、任意のレベルの水平又は垂直のウエーブレット変換の任意の時点のTT変換の代わりに、S変換のような別の可逆ウエーブレット変換フィルタを用いることができる。一実施例では、水平方向と垂直方向の両方にTT変換を利用して4レベル分解が実行される。一実施例では、4レベル分解において、4つのTT変換の中の2つがS変換で置き換えられる。これは圧縮の損失は少ないが、メモリ使用量に対する効果は大きい。水平方向の変換と垂直方向の変換は交互に適用されるであろう。

【0150】

なお、S変換とTT変換の組合せが水平方向及び垂直方向の変換を実施するために用いられてもよい。変換の順序は雑多でよいが、完全に可逆的であるためには、復号化器はその順序を知って逆の順序で逆操作を実行しなければならない。後述のように、復号化器は変換順序をヘッダで知らされるかもしれない。

【0151】

埋め込み順序付け

本発明では、ウエーブレット分解の結果として生成された係数はエントロピー符号化される。本発明においては、係数は最初に埋め込み符号化(embedded coding)を施されるが、この符号化では、視覚的に重要な順に係数が順序付けられ、または、より一般的に、何等かの誤差規準(例えば、歪み規準)を考慮して係数が順序付けられる。誤差または歪みの規準には、ピーク誤差と平均2乗誤差(MSE)が含まれる。また、ビット・シグニフィカンス空間配置(bit-significance spatial location)より、データベース照会のための妥当性を優先させるように、また方向別に(垂直、水平、斜め等)順序付けてもよい。

【0152】

データの順序付けは、符号ストリームの埋め込み量子化したものを生成するために行われる。本発明においては、2つの順序付け方法が用いられる。その一つは係数を順序付けするためのものであり、もう一つは係数中の2進値を順序付けするためのものである。本発明の順序付けは、ビットストリームを生成し、このビットストリームはその後にバイナリ・エントロピー・コードにより符号化される。

【0153】

タイル

本発明においては、変換と符号化の前に、画像はタイルに分割される。タイルは、完全独立に符号化される全体画像の部分画像であり、図11のように番号がつけられた画像上に配置された規則的な矩形格子により定義される。右端と下端のタイルは、原画像及びタイル・サイズに応じて色々なサイズになる。

【0154】

タイルは画像サイズ以下の任意の高さ、幅にしてよいが、タイル・サイズの選び方は性能に影響する。小さなタイル、特にラスト順画像の垂直方向の寸法が小さいタイルは、作業域用メモリを減らすことができる。しかし、タイルが小さすぎると、合図のための(sign

10

20

30

40

50

aling) オーバーヘッド、タイル境界での変換効率の損失、エントロピー・コードの立ち上がり適応という3つの要因により圧縮効率が下がる。タイルの寸法を、最も低い周波数成分の大きさの倍数(CREWツリー)、つまりレベル数の関数(2のレベル数乗)にするのが有利である。原画像のサイズによるが、 $128 \times 128$  又は  $256 \times 256$  のタイルが多くアプリケーションに適切と思われる。

【0155】

タイルは一連の画像の圧縮のために利用されるかもしれない。したがって、タイルされた画像は、時間的に異なる画像(映画のような)又は空間的に異なる画像(MRIの如き3D断面のような)かもしれない。これを知らせる格別の方法はないが、CMTが利用されるかもしれない。

10

【0156】

変換、コンテキスト・モデル、エントロピー符号化は1つの画像タイルの画素及び係数だけに作用する。それゆえに、画像タイルは順序によらずに解析又は復号化することができ、すなわちランダムにアドレスすることができ、あるいは、注目領域の伸長のため様々な歪みレベルに復号化することができる。

【0157】

1つの画像タイルの画素データは全部、符号化器で一度に利用できる、例えばメモリにバッファされる。ひとたび画素データが変換されれば、全ての係数データを水平コンテキスト・モデルに利用できる。全ての係数をランダムにアクセスできるので、画像タイル内部の埋め込みの順序は、符号化器及び復号化器が知っている限り任意でよい。エントロピー・コードは、この順序付けに関し無頓着であるから、その順序は圧縮率に大きな影響を及ぼすので注意して選ばねばならない。

20

【0158】

画像タイルは矩形配置されたツリー(LL係数及びその全ての子孫)の番号により定義される。各ツリー内の画素の数はウェーブレット分解のレベル数の関数である。

【0159】

画像基準格子は、各成分の大きさが格子点の整数倍である最小の格子面である。このことは、殆どの画像で、画像基準格子が最頻成分と同一であることを暗に意味する。

【0160】

1成分の画像又は全成分が同一サイズの画像の場合、画像基準格子は画像と同一サイズである(例えば格子点は画像の画素である)。複数の成分の全部が同一サイズというわけではない画像の場合、そのサイズは画像基準格子点の整数倍と定義される。例えば、CCIR601 YCrCb色成分系は、各Cr及びCb成分に対し2つのY成分を持つように定義される。したがって、Y成分は画像基準格子を定義し、Cr成分とCb成分はそれぞれ水平方向に2単位を、垂直方向に1単位をカバーする。

30

【0161】

ビット・シグニフィカンス表現

一実施例では、係数内の2進値に対し用いられる埋め込み順序はビットプレーン順である。係数はビット・シグニフィカンス表現で表される。ビット・シグニフィカンス表現は、最上位ビット(MSB)ではなくて符号(sign)ビットが、最初の非ゼロの絶対値ビットと共に符号化される符号・絶対値表現である。

40

【0162】

ビット・シグニフィカンス形式で表現される数には3種類のビット、すなわちヘッド(head)ビット、テール(tail)ビット及び符号(sign)ビットがある。ヘッドビットとは、MSBから最初の非ゼロ絶対値ビットまでの全てのゼロビットと、その最初の非ゼロ絶対値ビットである。その最初の非ゼロ絶対値ビットが存在するビットプレーンで、係数の重要性が定まる。最初の非ゼロ絶対値ビットの後からLSBまでのビットがテールビットである。符号ビットは符号(sign)を表わすにすぎない。MSBが非ゼロビットの数、例えば $\pm 2n$ は、ヘッドビットを1ビットしか持たない。ゼロの係数は、テールビットも符号ビットも持たない。図12にビット・シグニフィカンス表現の例を示す。

50

## 【 0 1 6 3 】

画素の輝度に関連して起こるような、値が非負整数の場合、採用し得る順序はビットプレーン順（例えば、最上位ビットプレーンから最下位ビットプレーンへの順）である。2の補数による負整数も許容される実施例では、符号ビットの埋め込み順序は、整数の絶対値の最初の非ゼロビットと同じである。したがって、1つの非ゼロビットが符号化されるまで、符号ビットは考慮されない。例えば、符号・絶対値表記法によれば、-7の16ビット数は

1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

である。ビットプレーン・ベースで、初めの12デシジョン(decision)は"無意味"すなわちゼロとなる。最初の1のビットは13番目のデシジョンに見つかる。次に符号ビット("負")が符号化される。符号ビットが符号化された後、テールビットが処理される。14番目と15番目のデシジョンは共に"1"である。

10

## 【 0 1 6 4 】

係数は最上位のビットプレーンから最下位のビットプレーンへと符号化されるので、データのビットプレーン数が正確にわからなければならない。本発明においては、データから計算される、又は画像の深度及びフィルタ係数から導き出される係数値の絶対値の上限を見つけることによって、ビットプレーン数が決定される。例えば、その上限が149のときには、有意な8ビットつまり8つのビットプレーンがある。ソフトウェアの速度のため、ビットプレーン符号化は用いられないかもしれない。別の実施例では、ビットプレーンが符号化されるのは、係数が2進数として意味をなす時だけである。

20

## 【 0 1 6 5 】

係数アラインメント(alignment)

本発明は、ビットプレーン符号化の前に係数相互のアラインメントを行う。これは、FFTやDCTと同様、異なった周波数サブバンド内の係数は異なった周波数を表すからである。本発明は、係数のアラインメントを行うことにより量子化を可能にする。量子化の重さが小さい係数ほど早いビットプレーン側へアラインメントされる(例えば左へシフトされる)。よって、ストリームが打ち切りされる場合、これらの係数は、それを定義するビットが、より重く量子化された係数に比べ多くなる。

## 【 0 1 6 6 】

一実施例では、係数はSNR又はMSEの見地から最高のレート・歪み性能が得られるようにアラインメントがなされる。MSEのような統計的誤差基準から見てほぼ最適のアラインメントを含め、多くのアラインメントが可能である。あるいは、アラインメントは係数データの物理視覚的(psychovisual)量子化を許すかもしれない。アラインメントは画像品質に(換言すればレート・歪み曲線に)相当な影響を及ぼすが、非損失性システムの最終的な圧縮率には殆ど影響しない。他のアラインメントは、特殊な量子化である注目領域忠実度符号化や解像度プログレッシブアラインメントに対応するかもしれない。

30

## 【 0 1 6 7 】

アラインメントは、圧縮データのヘッダで通知されるかもしれない。係数はビット・シグニフィカンス順に符号化されるが、最上位の重要性レベルは符号化単位中の係数から導き出される。各係数の符号ビットは、その係数が非ゼロの絶対値ビットを持つ最も上位の重要性レベルまで符号化されない。これは、絶対値がゼロの係数の符号ビットを符号化しないという利点がある。また、符号ビットは、埋め込み符号ストリーム中のそれが関連する点まで符号化されない。様々なサイズの係数に関するアラインメントは、符号化器及び復号化器のいずれも分かっているので、エントロピー・コードの効率に全く影響を与えない。

40

## 【 0 1 6 8 】

bビット/画素の画像の2レベルのTS変換及びTT変換分解における係数のビット深度を図13に示す。図14は、本発明における係数アラインメントに用いられる周波数帯域用乗数の例である。係数のアラインメントのために、1-HH係数のサイズが基準として用いられ、このサイズに対し相対的にシフトが与えられる。

50

## 【0169】

一実施例では、画像中の全ての係数のアライメントを生成するため、係数は最大の係数の絶対値を考慮してシフトされる。アライメント後の係数は、次に、重要性レベルと呼ばれるビットプレーン単位で、最上位の重要性レベル(MSIL)より最下位の重要性レベル(LSIL)へと処理される。符号(sign)ビットは、MSILの一部ではないので、各係数の最後のヘッドビットまで符号化されない。重要なことは、アライメントはエントロピー・コーダへビットが送られる順序を制御するに過ぎないことである。割増の0のビットのパディング、シフト、格納、符号化が実際に行われるわけではない。

## 【0170】

表1はアライメントの例を示す。

10

## 【0171】

## 【表1】

## 係数アライメント

1-HH 基準	1-HL,1-LH	2-HH	2-HL,2-LH	3-HH	3-HL,3-LH	4-HH	4-HL,4-LH
	左1	左1	左2	左2	左3	左3	左4

20

## 【0172】

様々なサイズの係数に関するアライメントは、符号化器と復号化器の両方に分かっているので、エントロピー・コーダの効率にはまったく影響を与えない。

## 【0173】

同じデータセットの符号化単位が異なったアライメントを持って構わないことに注意されたい。

## 【0174】

符号ストリームの順序付け

図15は、符号化ストリームの順序付けと符号化単位内における順序付けを示している。図15において、ヘッダ1001の後に、符号化単位1002が最も上の帯域より最も下の帯域へと順に続く。符号化単位の内部では、LL係数1003は符号化されずにラスタ(ライン)順に格納される。LL係数の後に、重要性レベルが、1ビットプレーンずつ、最上位のビットプレーンから最下位のビットプレーンへと順にエントロピー符号化される。この時、すべての係数の第1ビットプレーンが符号化され、次に第2ビットプレーンが符号化され、以下同様に符号化される。一実施例では、アライメントはヘッダ1001中に指定される。

30

## 【0175】

一実施例では、LL係数は、それが8ビット値のときは、符号化されずにラスタ順に格納されるにすぎない。LL係数のサイズが8ビット未満のときには、LL係数はパディングにより8ビットにされる。LL係数が8ビットより長いときには、LL係数は次のように格納される。まず、各係数の最上位の8ビットが符号化されずラスタ順に格納される。次に、係数の残りの下位ビットはパックされてラスタ順に格納される。例えば、10ビットのLL係数の場合、4個のLL係数の最下位ビットは1バイトにパックされる。このようにして、実際の画像の深度にかかわらず、係数毎に8ビットのLLデータを得られるので、簡略画像もしくはプレビュー画像の素早い生成が可能になる。

40

## 【0176】

各ビットプレーン期間内に係数が処理される順序は、低い解像度より高い解像度へ、かつ低い周波数より高い周波数への順である。各ビットプレーン内の係数サブバンドの順序は、高いレベル(低解像度、低周波数)より低いレベル(高分解能、高周波数)への順である。各周波数サブバンドの内部において、符号化はある決まった順序でなされる。一実施

50



例では、その順序はラスタ順、2 × 2 ブロック順、ジグザグ順、Peanoスキャン順、等々である。

【0177】

図15の符号ストリームを用いる4レベル分解の場合、その順序は次のとおりである。  
4-LL, 4-HL, 4-LH, 4-HH, 3-HL, 3-LH, 3-HH, 2-HL, 2-LH, 2-HH, 1-HL, 1-LH, 1-HH。

【0178】

符号ストリーム・データを重要性によって分けると、データを媒体に格納したり、ノイズのある通信路により伝送するのに有利である。データの異なった部分に、異なった冗長度を持つ誤り訂正/検出符号を用いることができる。最も高い冗長度の符号をLL係数のヘッダに用いることができる。(重要度を基準にして)重要性の低いエントロピー符号化データは、冗長度の低い誤り訂正/検出符号を用いることができる。訂正不可能な誤りが生じたときには、データの packets を廃棄(量子化)すべきか、または、通信路より再送信すべきか、もしくは記憶装置から再読みだしすべきかを決定するために、そのデータの重要性レベルも利用できる。例えば、高冗長度の誤り訂正 BCH 符号(Reed-Solomon符号など)が、ヘッダデータ、LLデータ、エントロピー符号化データの最も重要な4分の1のために用いられるかもしれない。エントロピー符号化データの残りの4分の3は、低冗長度の誤り検出チェックサム又はCRC(巡回冗長検査)によって保護されるであろう。一実施例では、BCH符号を用いる packets は常に再送信されて廃棄されず、一方、チェックサム又はCRC符号を持つ packets は再送信されず、データの送信を試みて失敗した後に廃棄されるであろう。

【0179】

一実施例においては、ヘッダデータがデータの各部分に使用される誤り訂正/検出符号を指定する。言い換えれば、ヘッダ中の情報が、誤り訂正符号を切り替えるべき時点を指示する。一実施例では、誤り訂正/検出符号は、通信路に利用される packets 間で、又は記憶媒体に利用されるブロック間で変更されるだけである。

【0180】

図26は、ヘッダ1901の後に符号化されないLL係数(1902)とエントロピー符号化データ1903が埋め込み順に続く符号ストリームを示す。図示のように、ヘッダ1901とLL係数1902は最高冗長度の符号を使用するが、エントロピー符号化データ1903は最低冗長度の符号を使用する。本発明は、最高の冗長度から最低の冗長度までの多くの異なった符号が使用されるスライディングスケールを採用することもできる。

【0181】

水平コンテキストモデル

本発明に利用される水平コンテキストモデルの一実施例を以下に説明する。このモデルは、係数の空間及びスペクトル従属性に基づいて符号化単位内のビットを利用する。隣接した係数及び親係数の利用可能な2進値を、コンテキストを生成するために使用してもよい。しかし、コンテキストはデコーダビリティを左右し、また、多少は効率的適応に影響を及ぼす。

【0182】

水平コンテキストモデルによる係数のモデリング

本発明は、バイナリ・エントロピー・コードのための埋め込みビット・シグニフィカンス順の係数により生成された符号ストリームをモデル化するためのコンテキスト・モデルを提供する。一実施例では、コンテキスト・モデルはランレングス・カウント(count)、空間モデル、符号(sign)ビット・モデル、テールビット・モデルからなる。ランレングス・カウントは、同じ状態のビットのランを測定する。空間モデルは、ヘッドビットに関する近傍係数及び親係数の情報を含んでいる。

【0183】

図16は、符号化単位の全ての係数の隣接係数を表す。図16において、隣接係数は分かりやすい地理的表記法で表されている(例えば、N=北、NE=北東、等々)。ある係数、例えば図16のPと、カレント・ビットプレーンが与えられたとすると、コンテキスト

10

20

30

40

50

モデルは、そのビットプレーンより前の符号化単位全てから得られるどの情報も利用することができる。本コンテキストモデルの場合、注目係数の親係数も利用される。

【0184】

水平ヘッドビット・コンテキストモデル

ヘッドビットは、最も圧縮できるデータである。したがって、圧縮率を上げるため、大量のコンテキストもしくは条件付けが使われる。隣接係数又は親係数の値を注目係数の注目ビットに対するコンテキストを決定するために利用するというよりもむしろ、その情報は本明細書においてテール情報と呼ぶ2ビットにまとめられる。この情報は、メモリに格納されてもよいし、隣接係数又は親係数から動的に計算されてもよい。テール情報は、最初の非ゼロの絶対値ビットがすでに見つかったか否か（例えば最初の”オン”ビットがすでに見つかったか否か）を示し、そして、すでに見ついているならば、幾つ前のビットプレーンであったかを示す。表2はテール情報ビットの説明である。

10

【0185】

【表2】

テール情報の定義

テール	定 義
0	オン・ビットは未だ現われていない
1	最初のオン・ビットは直前のビットプレーンにあった
2	最初のオンビットは2又は3ビットプレーン前にあった
3	最初のオンビットは4ビットプレーン以上前にあった

20

【0186】

この2ビットのテール情報から、そのテール情報がゼロか否かを示す1ビットの”テール・オン”ビットが合成される。一実施例では、テール情報とテール・オンビットは係数が符号化された直後に更新される。別の実施例では、その更新は、並列的コンテキスト生成を可能にするため、もっと後に行われる。

30

【0187】

さらに、この2ビットは、符号化される重要性レベルを示すために利用される。最初の2ビットプレーンは値0、第2の2ビットプレーンは値1、第3の2ビットプレーンは値2、残りのビットプレーンは値3を使用する。さらに、すべてゼロのヘッドビットのランゲス符号化がある。

【0188】

ヘッドビットのための10ビットのコンテキストは、親係数及びW係数それぞれの2ビット情報、N, E, SW, S各係数の1ビット情報、2ビットの重要性レベル情報からなる。

40

【0189】

一実施例では、一部又は全ての周波数帯域についてテール情報は用いられない。こうすることにより、周波数帯域を、その親を前もって復号化すねことなく復号化できるようになる。

【0190】

別の実施例では、各周波数帯域のビットプレーンの重要性レベルへの割り当てに、一つのアラインメントを用いる。親のテール・オン情報の決定に、もう一つのアラインメントを利用するが、これが利用する親のビットプレーンは実際に符号化されているビットプレーンより少ない。これにより、ある周波数帯域のいくつかのビットプレーンが、同じ重要性レベルの対応した親ビットプレーンを復号化せずに復号化できるようになる（図51参照

50

)。例えば、MSEアラインメントに基づいた親のテール・オン情報によらず、ピラミッド・アラインメントにより画像を符号化できる(図50参照)。これにより、復号化器は、MSEアラインメントを模擬し、又はピラミッド・アラインメントとMSEアラインメントの間の任意のアラインメントを模擬し、ピラミッド・アラインメントで復号化することができる。

【0191】

図29はコンテキスト従属関係を示す。子はその親に条件付けられる。したがって、親は、その子を復号化する前に復号化されなければならない、特に符号化時に用いたアラインメントと異なるアラインメントを用いて復号化する時にはそうである。

【0192】

水平符号(sign)ビット・コンテキストモデル

最後のヘッドビットの後で、符号ビットが符号化される。符号のコンテキストは、N係数が正であるか負であるか、符号がまだ符号化されていないかによって、3つ存在する。

【0193】

水平テールビット・コンテキストモデル

テールビットのためのコンテキストは、注目係数のテール情報の値によって3つ存在する。(テールビットを符号化しようとしているときには、テール情報値は1, 2又は3に限られることに注意されたい)

水平コンテキストモデルのためのステップ

システムのコンテキストモデルは、コンテキストを記述するために最高11ビットを使う。この数がまるまる指定されなくともよい。各ビット位置の意味は、前の2進値に依存する。まず第1に、ヘッドビットのある"ラン符号化"を提供するために、ただ1つのコンテキストが用いられる。ヘッドビットのランがないときには、各ビットは、あるコンテキストに寄与する隣接係数及び親係数により符号化される。ステップの具体例は以下のとおりである。

【0194】

1) ルックアヘッド(look-ahead)をすべきか判定する。

次のN個の係数とそれらの北側に隣接する係数のテール情報がすべてゼロならば、システムはステップ2に進む。そうでなければ、次のN個の係数のためステップ3に進む。一実施例では、N=16である。

【0195】

2) ルックアヘッド手順を行う。

次のN個の係数の符号化すべきカレント・ビットプレーンのビットがゼロならば、1つの0が符号化され、システムはステップ1から次のN係数に移行する。そうでなければ、1つの1が符号化され、システムは次のN係数のためステップ3に進む。

【0196】

3) 注目係数の状態を判定し符号化する。

注目係数のテール情報が0ならば、注目係数のカレント・ビットプレーンのビットは、西の係数と親係数(オプション)のテール情報の2ビット、北西、東、南西、南の係数のテールオンビット、及び重要性レベル情報の2ビットにより作られる1024個の可能なコンテキストにより符号化され、システムはステップ4に進む。なお、一実施例では、親係数は利用されないため、コンテキストは隣接係数と重要性レベル情報だけから作られる。注目係数のテール情報が0でなければ、注目係数のカレント・ビットプレーンのビットはテールビットであり、注目係数のテール情報の2ビットから作られる3つのコンテキストにより符号化される。

【0197】

4) カレント・ヘッドビットの状態を判定し、必要なら符号(sign)ビットを符号化する。

注目係数のカレント・ビットプレーンのビットが1ならば、注目係数の符号(sign)は、北係数のテールオンビット及び符号ビットにより作られる3つの可能なコンテキストによ

10

20

30

40

50

り符号化される。

【0198】

図17は前述のプロセスのフローチャートである。図17において、白いブロックは符号化と関係がなく、黒いブロックは符号化に関係がある。図に示さないが、各エントロピー符号化デシジョン(decision)に対し1つのコンテキストが定義される。前述の作用及びフローは、当業者には理解できるであろう。

【0199】

水平コンテキストモデルの一例が、入力係数を符号/絶対値形式に変換する符号/絶対値ユニットの一例とともに、米国特許出願第08/498,695号(1995年6月30日受理、"Method and Apparatus For Compression Using Reversible Wavelet Transforms and an Embedded Codestream")、及び、米国特許出願第08/498,036号(1995年6月30日受理、"Reversible Wavelet Transform and Embedded Codestream Manipulation")に記述されている。

10

【0200】

エントロピー符号化

一実施例では、本発明により実施されるエントロピー符号化は、バイナリ・エントロピー・コードによって実行される。一実施例では、エントロピー・コード106はQコード、QMコード、有限状態マシン、又は高速並列コード等からなる。単一のコードを用いて単一の出力符号ストリームを生成してもよい。あるいは、複数の(物理又は仮想)コードを用い、複数の(物理又は仮想)データストリームを生成してもよい。

20

【0201】

一実施例では、本発明のバイナリ・エントロピー・コードはQコードからなる。Qコードに関する情報を得るには、Pennebaker, W. B., et al., "An Overview of the Basic Principles of the Q-coder Adaptive Binary Arithmetic," IBM Journal of Research and Development, Vol.32, pg. 717-26, 1988 を読みたい。別の実施例では、バイナリ・エントロピー・コードは、周知の効率的なバイナリ・エントロピー・コードであるQMコードを用いる。QMコードは、確率スキューが非常に高いビットに対し特に効率的である。QMコードはJPEG規格とJBIG規格の両方で利用される。

【0202】

バイナリ・エントロピー・コードは、有限状態マシン(FSM)コードでもよい。このようなコードは、確率及び事象(outcome)から圧縮ビットストリームへの単純な変換を提供する。一実施例では、有限状態マシン・コードは、符号化器、復号化器の両方として、テーブルルックアップにより実現される。多様な確率予測法を、このような有限状態マシンコードに利用できる。0.5に近い確率に対する圧縮率が非常によい。大きくスキューした確率に対する圧縮率は、用いられるルックアップテーブルのサイズに依存する。QMコードと同様、有限状態マシン・コードは、デシジョンが発生順に符号化されるので、埋め込みビットストリームに有効である。出力はルックアップテーブルにより決められるので、"キャリーオーバー"(carry over)問題が起こる心配は全くない。実際には、QコードやQMコードと違って、符号化と圧縮出力ビットの生成との間に最大の遅延がある。一実施例では、本発明の有限状態マシン・コードは、1993年12月21日発行の米国特許第5,272,478号"Method and Apparatus for Entropy Coding"に述べられているBコードからなる。

30

40

【0203】

一実施例では、本発明のバイナリ・エントロピー・コードは高速並列コードからなる。QMコードもFSMコードも、1ビットずつ符号化又は復号化される必要がある。高速並列コードは、数ビットを並列に処理する。一実施例では、高速並列コードは、圧縮性能を犠牲にすることなく、VLSIハードウェア又はマルチプロセッサ・コンピュータで実現される。本発明において利用し得る高速並列コードの一例が、1995年1月10日発行の米国特許第5,381,145号"Method and Apparatus for Parallel Decoding and Encoding of Data"に述べられている。

50

## 【0204】

殆どの効率的なバイナリ・エントロピー・コードは、基本フィードバックループによって速度が制限される。考えられる一解決法は、入力データストリームを複数のストリームに分割して並列の複数の符号化器に与えることである。それら符号化器の出力は、複数の可変長符号化データ・ストリームである。この種の方法の一つの課題は、データを単一のチャンネルでどのようにして伝送するかである。米国特許第5,381,145号に述べられている高速並列コードは、この課題を、それら符号化データストリームをインターリーブする方法によって解決する。

## 【0205】

本発明において利用されるコンテキストの多くは定確率であり、このことがBコードのような有限状態マシン・コードを特に有効なものにする。なお、システムが0.5に近い確率を利用する場合、上記特許に開示された高速並列コード及び有限状態マシン・コードは共にQコードより効率よく動作する。よって、それら両方のコードは、本発明のコンテキストモデルに対し本質的な圧縮上の強みを持っている。

10

## 【0206】

別の実施例では、バイナリ・エントロピー・コード及び高速m元コードの両方が利用される。高速m元コードはハフマン・コードでよい。

## 【0207】

本発明の符号化及び復号化のプロセス

図18乃至図20のフローチャートは、本発明の符号化プロセス及び復号化プロセスの例を表している。処理ロジックは、ソフトウェア及び/又はハードウェアによって実現してよい。

20

## 【0208】

図18は本発明の符号化プロセスの一例を示す。図18において、符号化プロセスの最初で、処理ロジックが1タイル分の入力データを取得する(処理ブロック1201)。

## 【0209】

処理ロジックは次に、バイナリ符号化を実行しなければならないか判定する(処理ブロック1202)。バイナリ符号化を実行すべきときには、プロセスは処理ブロック1211に進み、処理ロジックは入力データに対しGray符号化を実行し、そして各係数の各ビットをバイナリ方式コンテキストモデルによりモデル化する(処理ブロック1212)。

30

処理は処理ブロック1208へ進む。

## 【0210】

バイナリ符号化を実行する必要がなければ、プロセスは処理ブロック1203へ進み、処理ロジックはデータに可逆フィルタをかける。可逆フィルタをかけた後、処理ロジックは別の分解レベルが必要か判定する(処理ブロック1204)。別の分解レベルが必要ならば、処理ロジックはLL係数に可逆フィルタをかけ(処理ブロック1205)、そして処理は処理ブロック1204へ戻り再び判定を行う。別の分解レベルが必要でないならば、プロセスは処理ブロック1206へ進み、処理ロジックは係数を符号・絶対値形式へ変換する。それから、処理ロジックは各係数の各ビットを水平コンテキストモデルによりモデル化し(処理ブロック1207)、プロセスは処理ブロック1208へ進む。

40

## 【0211】

処理ブロック1208で、処理ロジックは各係数の各ビットを符号化する。そして、処理ブロックは各符号化データを送信又は格納する(処理ブロック1209)。

## 【0212】

処理ブロックは、次に、画像中に他にもタイルが使われているか判定する(処理ブロック1210)。画像中に他のタイルがあるならば、処理ロジックは処理ブロック1201へループバックし処理が繰り返されるが、他のタイルがなければプロセスは終了する。

## 【0213】

図19は、本発明の復号化プロセスの一例を示す。図19において、プロセスはまず1タイル分の符号化データを取得する(処理ブロック1301)。次に、処理ロジックはその

50

符号化データをエントロピー復号化する（処理ブロック1302）。そして、処理ロジックは、そのデータがバイナリ復号化されなければならないか判定する（処理ブロック1203）。そのデータがビット毎にバイナリ復号化されなければならないときには、プロセスは処理ブロック1311へ進み、処理ロジックは各係数の各ビットをバイナリ方式コンテキストモデルによりモデル化し、そして、そのデータに対し逆Gray符号化を施す（処理ブロック1312）。この逆Gray符号化の後、プロセスは処理ブロック1309へ進む。

【0214】

バイナリ復号化が実行される必要がないときには、プロセスは処理ブロック1304へ進み、処理ロジックは各係数の各ビットを水平コンテキストモデルによりモデル化する。そして、処理ロジックは各係数をフィルタ処理に適した形式へ変換し（処理ブロック1305）、係数に可逆フィルタをかける（処理ブロック1306）。 10

【0215】

可逆フィルタをかけた後、処理ロジックは別のレベルの分解があるか判定する（処理ブロック1307）。別レベルの分解があるならば、プロセスは処理ブロック1308へ進み、処理ロジックは係数に可逆フィルタをかけ、そしてプロセスは処理ブロック1307へループバックする。別の分解レベルが必要でなければ、プロセスは処理ブロック1309へ進み、再構成されたデータは送信されるか格納される。

【0216】

次に、処理ロジックは画像中に他にタイルがあるか判定する（処理ブロック1310）。画像中にほかにタイルがあるときには、処理は処理ブロック1301へループバックして処理が繰り返されが、他にタイルがなければ、プロセスは終了する。 20

【0217】

図20は本発明によるビット・モデル化のためのプロセスの一例を示す。図20において、ビット・モデル化プロセスは初めに係数変数Cを最初の係数に設定する（処理ブロック1401）。つぎに、 $|c| > 2^s$  の判定を行う（処理ブロック1402）。判定結果がyesのときには、処理は処理ブロック1403に進み、処理ロジックはテールビット用モデルを用いて係数CのビットSを符号化し、そして処理ブロック1408に処理が進む。このテールビット用モデルは静的（非適応型）モデルでかまわない。 $|c|$ が $2^s$ より大きくないときには、処理は処理ブロック1404に進み、処理ロジックはテンプレートをヘッドビット（頭の0と最初の"1"ビット）に適用する。テンプレートを適用した後、処理ロジックは係数CのビットSを符号化する（処理ブロック1405）。可能なテンプレートを図21に示す。 30

【0218】

次に、係数CのビットSがオンであるか判定する（処理ブロック1406）。係数CのビットSがオンでなければ、処理ブロック1408へ進む。一方、係数CのビットSがオンならば、処理は処理ブロック1407に進み、処理ロジックは符号(sign)ビットを符号化する。そして処理は処理ブロック1408へ進む。

【0219】

処理ブロック1408で、係数Cが最後の係数であるか判定する。係数Cが最後の係数でなければ、処理は処理ブロック1409に進み、係数変数Cは次の係数に設定され、そして処理ブロック1402から処理を続ける。一方、係数Cが最後の係数ならば、処理ブロック1410に進み、Sが最後のビットプレーンであるか判定する。Sが最後のビットプレーンでなければ、ビットプレーン変数Sが1だけデクリメントされ（処理ブロック1411）、処理ブロック1401から処理を続ける。Sが最後のビットプレーンならば、処理は終了する。 40

【0220】

TS変換設計

本発明は、一実施例において、バッファメモリ内の適所においてTS変換を計算する。これを行う際、計算値を再配置するためのメモリの余分なライン及び余分な時間は必要とさ 50

れない。TS変換については既に述べたが、本発明は、臨界サンプリングしたものに対しオーバーラップ変換を適用する。別の実施例では、TT変換が用いられる。

【0221】

図24(A)乃至図24(C)は、本発明の変換の計算中に本発明により採用されるメモリ操作方法を示す。図24(A)はメモリの初期状態を示す。図24(A)において、メモリの最初の行には、前の値(n-1)の平滑("S")係数及び詳細("D")係数(計算済み)、カレント値(n)の平滑("S")係数及び部分的に完成した詳細係数("B")が、4つの入力標本("X")の値( $X_{2n+2}$ ,  $X_{2n+2}$ ,  $X_{2n+4}$ ,  $X_{2n+5}$ )とともに入っている。変換計算の中間結果は、図24(B)の同じメモリ行に示されている。この行の変更点は、第5記憶エレメント及び第6記憶エレメントにおいて、 $X_{2n+2}$ と $X_{2n+3}$ の値が $S_{n+1}$ と $B_{n+1}$ の値に置き換わったことだけであることに注意されたい。このように、もはや必要でない格納値を変換計算中に生成された結果によって置き換えることにより、本発明はメモリスペースを節約する。図24(C)は変換が完了し詳細出力 $D_n$ を生成後における同じメモリ行を示す。図24(B)から唯一変わった点は部分的に完成した詳細係数 $B_n$ が詳細出力 $D_n$ で置き換わったことである。

10

【0222】

nに関し詳細出力が計算された後、変換計算プロセスは、詳細出力 $D_{n+1}$ を計算するために行を下げて計算を続ける。

【0223】

次に示すコード例を、変換を行うため用いてよい。フォワード変換とリバース変換のための水平コードが含まれていることに注意されたい。

20

【0224】

以下において、変数 $s_{oo}$ は $S_{n-1}$ 値を指し、変数 $o_{so}$ は $S_n$ 値を指し、変数 $o_{os}$ は $S_{n+1}$ を指す。

【0225】

フォワードTS変換の一例のためのコード(C言語)の具体例は以下の通りである。

```
/*
TSFoward_1()
*/

void TSForward_1(long *x, int width)
{
    long *start=x;
    long *ox=x+2;
    long soo;
    long oso;
    long oos;

    oso>(*x+*(x+1))>>1;
    oos>(*ox+*(ox+1))>>1;
    soo=oso;

    *(x+1)=*x-*(x+1);
    *x=oso;

    while((ox+2)-start<width){
        x=ox;
        os+=2;

        soo=oso;
        oso=oos;

        oos>(*ox+*(ox+1))>>1;
    }
}
```

10

20

30

40



```

        *(x+1)=*x-*(x+1)+((oos-soo+2))>>2);
        *x=oso;
    }

x=ox;

soo=oso;
oso=oos;

oos-soo;

*(x+1)=*x-*(x+1);
*x=oso;
}

```

【 0 2 2 6 】

インバースTS変換の一例のためのコード(C言語)の具体例は以下のとおりである。

/\*

TSReverse\_1()

\*/

```

void TSReverse_1(long *x, int width)
{
    long *start=x;
    long *d=x+1;
    long ns=*(x+1);
    long P;

```

```
while(x+2-start<width){
```

```
    p=*d-((*x+2)-ns+2)>>2);
```

```
    ns=*x;
```

```
    *d=*x-(P)>>1);
```

```
    *x+=((p+1)>>1);
```

10

```
    x+=2;
```

```
    d=x+1;
```

```
}
```

```
p=*d;
```

```
*d=*x-(p)>>1);
```

20

```
*x+=((p+1)>>1);
```

#### 【 0 2 2 7 】

1次元の例のみ示したが、本発明は多次元及び多レベルにも利用し得る。なお、この手法は、他の計算のためには必要でなくなった値を部分的又は最終的結果により1対1に置き換える任意のオーバーラップ変換に利用できる。

#### 【 0 2 2 8 】

図25は、3レベル用のメモリバッファの2次元表現を示す。図25において、各ブロック1801～1804のメモリロケーションには係数値が入っている。すなわち、ブロック1801～1804のそれぞれは、係数値の8×8ブロックである。

30

#### 【 0 2 2 9 】

係数は、2の自然数乗の間隔で配置される。レベルと、S又はDのオフセットが与えられれば、どの係数もアクセスできる。それゆえに、特定のレベルと水平及び垂直周波数を選択することによって、アクセスをすることができる。バッファはラスタ順にアクセスされてもよい。

#### 【 0 2 3 0 】

ユニット・バッファ構成

本発明の一実施例においては、単一のバッファで圧縮システムの変換ブロック、コンテキストモデル・ブロック、符号化ブロックをサポートする。このバッファは、係数を効率的にアクセスでき他にメモリを必要としない2次元のスクローリング・メモリバッファである。バッファの各ラインはライン・アクセス・バッファに格納されたポインタを介してアクセスされる。図23(A)及び図23(B)は、このスクローリング・バッファの構成を示し、ライン・アクセス・バッファ1601にバッファ1602の各ラインを指すポインタが入っている。

40

#### 【 0 2 3 1 】

スクロールは、ライン・アクセス・バッファに格納されるポインタを並べ直すことによりなされる。その例が図23(A)及び図23(B)に示されている。図23(A)はバッファの初期状態を示す。図23(B)を見るに、ラインA, B, Cがバッファから取り除かれ、ラインG, H, Iによってそれぞれで置き換えられた後に、バッファにスクローリング・バッファの作用を付与するため、ライン・アクセス・バッファのポインタは、第1

50

のポインタがバッファ内のラインDを指し、第2のポインタがラインEを指し、第3のポインタがラインFを指すように変更される。次に、ラインG, H, Iを指すポインタがライン・アクセス・バッファの最後の3つの位置を占める。なお、6ラインのバッファを持つことに本発明が限定されるわけではなく、これはあくまで一例として用いられるにすぎない。もっとライン数の多いバッファが一般的に使用されるが、これは当業者には周知であろう。しかして、ライン・アクセス・バッファ経由のアクセスは、記憶を物理的に移動させることなく、ユニット・バッファがスクロールするような外観を呈する。これにより、スピードを犠牲にせずに、最小限のメモリを使用できるようになる。

**【0232】**

本発明においては、そのようなユニット・バッファを用いることにより、常に画像の1つの帯域のみメモリに記憶しつつ、画像全体に対するオーバーラップ変換の適用をサポートする。これを達成するため、少なくとも1バンドのウェーブレット・ユニットを構成するウェーブレット係数のセットを完全に計算するために必要な数分の画像のラインに対してだけウェーブレット変換を適用する。このような場合には、計算の完了したウェーブレット係数のセットを、モデル化し、エントロピー符号化し、そしてウェーブレット・ユニット・バッファの該当部分より取り除くことができる。計算が未完のウェーブレット係数は、次の反復で計算を完了させるために、そのまま残る。そして、ライン・ポインタを並べ替えることによりウェーブレット・ユニット・バッファをスクロールし、ウェーブレット・ユニット・バッファの空き部分にほかのデータを入れることができる。これで、計算途中のウェーブレット係数を完全に計算することができる。

**【0233】**

一例として、ハイパスフィルタがカレント係数及び次のローパスフィルタ係数に依存するオーバーラップ変換を適用することを考える。この例の場合、たった2レベルの分解が画像データに適用されるであろうが、これは1つのウェーブレット・ユニットが4エレメント長であることを暗に示す。

**【0234】**

少なくとも1つの帯域のウェーブレット・ユニットを構成するウェーブレット係数のセットを完全に計算するために、ウェーブレット・ユニット・バッファの高さは少なくとも8ラインつまり2ウェーブレット・ユニットである。

**【0235】**

2次元のウェーブレット・ユニット・バッファに対しウェーブレット変換を適用する際には、まず1次元のウェーブレット変換がバッファの各行(ライン)に対し適用される。それから、1次元のウェーブレット変換がバッファの各列に対し適用される。

**【0236】**

ウェーブレット・ユニット・バッファの各列に対し1次元ウェーブレット変換を適用する時に、ユニット・バッファに格納されていない画像のエレメントに依存する各列の最後のエレメントについては、ハイパスフィルタの計算を部分的にしか終わらせることができない。これが図55に示されている。

**【0237】**

第2レベルのウェーブレット分解を実行する時に、再び、各列の最後のエレメントについては、ハイパスフィルタの計算を部分的にしか終わらせることができない。これが図56に示されている。

**【0238】**

なお、一実施例においては、多くの分解レベルを利用する時に、SS係数(第2分解レベルの場合は図55の1SS、第3レベルの場合は図56の2SS)にのみウェーブレット変換が適用されるだろう。このような場合、ユニット・バッファの行及び列のロケーションは、バッファの適切なエントリーの読み書きを保証するようスキップされるだろう。

**【0239】**

この例では、バッファの上半分には計算が完了したウェーブレット係数のセットが入っており、このウェーブレット係数セットは1帯域のウェーブレット・ユニットを構成してお

10

20

30

40

50

り、モデル化し、エントロピー符号化し、そしてバッファから取り除くことができる。

【0240】

バッファの上半分が空くと、バッファを高さの半分だけスクロールすることができる。ここで、画像の次の4ラインをバッファに読み込むことができる。バッファに格納された新たなラインそれぞれに対し、1次元ウェーブレット変換を適用することができる。バッファの列方向に、部分的に計算された係数を完全に計算することができるが、再び、各列の最後のエレメントは部分的にしか計算することができない。

【0241】

第2レベルのウェーブレット分解のためにも同様のことが行われる。再び、バッファの上半分は計算の完了したウェーブレット係数が入り、その段階でプロセスはほかに処理すべき画像のラインがなくなるまで反復する。

10

【0242】

ライン・アクセス・バッファ内のライン・ポインタの並べ替えは、様々なやり方で行うことができる。一つのやり方は、新しいライン・アクセス・バッファを作り、それに旧ライン・アクセス・バッファからポインタをコピーする方法である。ウェーブレット・ユニット・バッファの高さを法としてスクロールするためには、旧ライン・アクセス・バッファのエレメント  $i$  に格納されているポインタが、インデックス ( $i + \text{ライン数}$ ) にコピーされることになる。

【0243】

なお、圧縮システムの3つのステージは、バッファ内のデータに対し、そのデータがバッファより出される前に遂行されるので、係数は一般的に様々に順序つけられること留意すべきである。ラスタ順データ操作が行われる場合には、本発明のスクロール・バッファは最小限のメモリで間に合う。

20

【0244】

ソフトウェア（及び/又はハードウェア）によりライン・アクセス・バッファを管理してポインタを操作する。このソフトウェアも、バッファ内のどのデータが処理を完了しバッファから出してよいか知っている。

【0245】

アラインメント法

本発明は、係数値を左へ任意量だけシフトする。一実施例では、このアラインメントは、仮想的アラインメント法によって行われる。この仮想的アラインメント法は、係数を実際にシフトしない。その代わりとして、係数を1ビットプレーンずつ処理する間に、特定の係数についてアラインメントが必要とされる実際のビットプレーンが計算される。重要性レベルと特定の係数に対し適用されるべきシフト量とが与えられれば、本発明は、その係数の所望の絶対ビットプレーンを、それが可能なビットプレーンの範囲内であればアクセスする。すなわち、特定の係数の所望の絶対ビットプレーンは、カレント重要性レベルから該係数に適用されるべきシフト量を差し引いたものにより与えられる。この所望ビットプレーンは、それが最小の有効な絶対ビットプレーンより大きいか又は等しく、かつ最大の有効な絶対ビットプレーンより小さいか又は等しいときに有効とみなされる。

30

【0246】

2つのアラインメント法が普通である。その中の第1の方法は、平均2乗誤差(MSE)法と呼ばれるもので、フルフレーム(full-frame)の再構成画像を原画像と比較した時にMSEが縮小もしくは最小化されるように、係数をアラインメントする方法である。図50は、このアラインメントの一例である。図51も参照のこと。

40

【0247】

第2の方法は、ピラミッド型のアラインメント法であり、画像がピラミッド・レベルのサイズに再構成される場合に良好なレート・歪み性能を提供する。ここでは、隣接レベルの係数は共通の重要レベルを持たない、つまりオーバーラップがない。図51の左側にあるアラインメントは、3レベルTS変換のための厳密にピラミッド型のアラインメントを表す。図51の右側は、レベル2のピラミッド型アラインメントを表す。(図51の厳密に

50

ピラミッド型の部分は、レベル3と2のピラミッド型アラインメントと呼んでよからう。

) 各場合において、レベル内部の係数はMSEに関連してアラインメントされる。

【0248】

図52は、メモリ記憶係数と一つのアラインメントとの間の典型的な関係を示す。

【0249】

本発明によれば、実際のシフト操作を行う必要がないため、メモリサイズの制約がなくなる。さらに、本発明は、余分なメモリを必要とせず、任意のアラインメント法を簡単に実施できる。

【0250】

ヒストグラム圧縮

本発明はヒストグラム圧縮を利用してよい。一実施例では、変換又はバイナリ方式の処理を施される前にヒストグラム圧縮が利用される。ヒストグラム圧縮は、一部の画像に対する圧縮率を向上させる。そのような画像は通常、ダイナミックレンジの一部の値がどの画素にも使われない画像である。言い換えれば、画像のダイナミックレンジにギャップが存在する。例えば、ある画像が合計256の値の中の0と255の値しかとらないときには、その原画像と一対一対応を持つが、ダイナミックレンジのずっと小さな新たな画像を生成できる。この画像生成は、整数を画像のとり値に写像する増加関数を定義することにより達成される。例えば、画像が0と255の値しか使わないときには、マッピングで0を0に、1を255に写像する。別の実施例では、画像が偶数(又は奇数)画素しか持たないときには、画素値を0~128の値に再写像する。

【0251】

ヒストグラム圧縮を行った後に、画像データに本発明の可逆埋め込みウェーブレットによる圧縮を施してよい。このように、ヒストグラム圧縮は前処理モードで用いられる。一実施例では、ヒストグラムは、ブーリアン(Boolean)ヒストグラムを基礎としており、値が生じるか否かのリストを保持するようなものである。まず、全ての生起数が昇順に記録される。次に、それぞれの値は0から順に写像される。

【0252】

一実施例では、誤差の影響を減らすためガード(guard)画素値が用いられる。隣接した再写像画素値が大きなギャップで隔てられた実画素値に対応するかもしれないので、再写像値の小さな誤差が実値の大きな誤差をもたらすことがある。再写像値の近くに割増値を追加することにより、そのような誤差の影響は減少するであろう。

【0253】

原画像を再構成するために、マッピングが利用されたことが復号化器へ通知される。このマッピングをヘッダ中に指示してもよい。これにより、復号化器において後処理のために同様のテーブルを作成できるようになる。一実施例では、復号化器はレンジをタイル毎に通知される。一実施例では、本発明は、まず当該マッピングが行われることを知らせ、次に欠落した値(例えば上例の254)の数を知らせる。ヒストグラム圧縮の利用の有無を知らせるためのコストは、たったの1ビットである。このビットの後に、全ての再写像値のテーブルが続くことになろう。

【0254】

一実施例では、1タイルずつヒストグラム圧縮を実行する時の通知量を減らすため、1つのビットで、新しいブーリアン・ヒストグラムが直前に使用したブーリアン・ヒストグラムと同じか違うかを知らせる。このような場合、新たなブーリアン・ヒストグラムが復号化器に通知されるのは、それが直前のヒストグラムと異なるとき(のみ)である。新しいブーリアン・ヒストグラムが前のものと違う場合でも、その間に類似点があるのが普通である。より詳しくいえば、2つのヒストグラムの排他的論理和はエントロピー・コードによる圧縮性がより高いので、それを生成して復号化器へ通知してもよい。

【0255】

ヒストグラムは、サイズのダイナミックレンジと同じくらいのビット数(例えば8ビット深度の場合は256ビット)を送信することにより通知できる。そのビットの並び順は画

10

20

30

40

50

素値に対応する。この場合、あるビットが1であるのは、それに対応した値が画像中に用いられるときである。ヘッダのコストを削減もしくは最小化するため、このビット列を、第1次マルコフ(Markov)コンテキストモデルのもとでエントロピー符号化してもよい。

【0256】

別の実施例では、欠落した値が過半数のときには発生した値が順に記録されるであろうが、そうでないときには欠落した値が順に記録される。

【0257】

一実施例では、本発明のバイナリ方式が、パレット化された画像の圧縮のために利用されるであろう。パレットはヘッダに格納されるであろう。しかし、パレット化画像は埋め込まれないであろうから、損失性伸長のための量子化は妥当な結果をもたらさない。別の実施例では、パレット化画像は連続階調(カラー又はグレースケール)画像に変換され、各成分が変換方式又はバイナリ方式により圧縮されるかもしれない。これは妥当な損失性圧縮が可能である。

10

【0258】

ある種の画像は、1つの指定色(又は指定色の小サブセット)が特定の目的のために使用された連続階調画像である。その特定目的色は注記用かもしれない。例えば、グレースケールの医用画像には、その画像を識別するためのコンピュータ生成のカラー文字があるかもしれない。もう一つの特定目的色は、オーバーレイ画像において画素が透過であって、下側の画像の画素が代わりに表示されることを示すかもしれない。禁止された色は別の成分画像に分解されるであろう。そして、連続階調成分及び特殊色成分は、変換方式又はバイナリ方式で圧縮/伸長されるであろう。

20

【0259】

なお、変換方式及びバイナリ方式は輝度データのために利用されることが多いが、アルファ混合用のアルファ・チャンネルのような他の形式の2次元データが用いられてもよい。

【0260】

パーサ(parser)

本発明によれば、符号ストリームを、伸長することなく、送信又は復号化の前に構文解析できるようになる。この構文解析は、ビットストリームを打ち切り、特定の量子化のために必要な量の情報だけを伝達することができるパーサによって実行される。このパーサを支援するため、マーカー(marker)及びポインタがビットストリーム内の符号化単位の各ビットプレーンの位置を決定する。

30

【0261】

本発明は、画像圧縮システムにおいて構文解析によって実施される装置依存の量子化を提供する。圧縮システムにおいてマーカーを用いることにより、符号化後に装置に応じ選択される量子化が可能になる。出力装置はその特性をパーサに報告し、パーサはその特定装置向けに符号化済みファイルを量子化する。この量子化は、ファイルの一部を省くことによる。可逆ウェーブレット変換の利用は、画像の非損失復元、あるいは、装置に応じた視覚的に非損失の色々な歪みでの画像の復元を可能にする。

【0262】

本発明によれば、量子化を符号化後に実行することが可能になる。図27及び図28は、パーサを備えた圧縮システムのブロック図である。図27及び図28において、元の圧縮されていない画像2101が本発明の圧縮装置2102に入力される。圧縮装置2102は、画像2101を圧縮ビットストリーム2103へ非損失圧縮するとともに、圧縮ビットストリーム2103にマーカーを付加する。

40

【0263】

圧縮ビットストリーム2103はパーサ2104に入力し、パーサ2104は圧縮ビットストリーム2103のある部分を出力として提供する。そのある部分は、圧縮ビットストリーム2103の全部であるかもしれないし、その一部分だけかもしれない。要求側のエージェントもしくは装置は、伸長画像が必要とされる時に、その装置特性をパーサ2104に与える。それに応じて、パーサ2104は圧縮ビットストリーム2104の適切な部

50

分を選択して送出する。パーサ2104は画素もしくは係数レベルの演算もエントロピー符号化/復号化も行わない。別の実施例では、パーサ2104はそのような働きを多少は果たすかもしれない。

【0264】

パーサ2104は、低解像度用の圧縮係数を選択することにより、モニタに画像を表示するための符号化データを提供することができる。それと異なる要求に対しては、パーサ2104は注目領域(ROI)の非損失性伸長を可能にするように圧縮データを選択する。一実施例では、パーサ2104は、要求に従って、プレビュー画像からプリンタ解像度画像又はフルサイズの医用モニタ画像(恐らく16ビットの画素深度を持つ)までの変化に必要なビットを送出する。

10

【0265】

パーサ2104により提供されたデータは通信路及び/又は記憶装置2106へ出力される。伸長装置2107は、そのデータにアクセスし、圧縮データを伸長する。伸長された、すなわち再構成されたデータは、伸長画像2108として出力される。

【0266】

図29において、2HH周波数帯域内のビットプレーンは3HH周波数帯域からの情報を利用して符号化される。ビットプレーンをより明瞭に示すため、図50及び図51に図29が書き直されている。図50のように係数が格納される場合(MSE)には、圧縮ビットストリームの打ち切りはMSEレート・歪み最適量子化とほとんど同一である。この打ち切りが、図50に陰をつけたマーカーで示されている。図30を調べると、この配列はプリンタには適切かもしれないが、モニタには十分でないであろう。図51に示すように、係数が"ピラミッド状に"格納される場合、つまり、ある周波数帯域の全ビットが最初に格納される場合には、ビットストリームの打ち切りは様々な解像度の画像を提供する。

20

【0267】

マーカーを巧みに使えば、両タイプの打ち切りが可能になり、解像度の低い、忠実度の低い画像を発生するであろう。図50及び図51中の濃淡の変化は、最高解像度で低忠実度のビットストリームを発生するであろうビットストリームの打ち切りを表す。LH, HL, HH係数の全部をさらに打ち切れれば、画像の解像度を下げることになる。

【0268】

多くの画像圧縮の応用では、画像は、一度だけ圧縮されるが、何度も伸長されるかもしれない。あいにく、たいていの圧縮システムは、許容される損失量と適当な量子化とが符号化の時点で決定されなければならない。プログレッシブ・システムは段々に精細になる一揃いの画像を与えるが、非損失性再構成は一般にできない、すなわちプログレッシブ・ビルドアップと関係のない非損失な方法で符号化された"差分画像"を送り出すことにより、非損失性再構成が提供される。

30

【0269】

本発明においては、符号化器は、種々の係数を周波数及びビットプレーン要素に分解するに足るだけの情報を保存する。一実施例では、次のエントロピー符号化データ単位に何が入っているか知らせるためのマーカーがビットストリームに挿入される。例えば、マーカーは、次のエントロピー符号化データ単位に、最上位より3番目のビットプレーンのためのHH周波数情報が入っていることを示すかもしれない。

40

【0270】

誰かがモニタ上で画像を調べたいとすると、低解像度のグレースケール画像を生成するために必要な情報を要求するであろう。そのユーザがその画像を印刷したいときには、高解像度の2値画像を生成するために必要な情報が要求されるであろう。最後に、そのユーザが圧縮試験を実施したいとき、あるいはセンサ雑音の統計分析や医療診断を行いたいときには、その画像の非損失版が要求されるであろう。

【0271】

図31は、パーサ、復号化器、出力装置とのやりとりに関するブロック図である。図31において、パーサ2402は、マーカー付の非損失性圧縮データ、並びに1つ以上の出力

50

装置、例えば図示のディスプレイ・モジュール 2 4 0 5 の装置特性を受け取るように接続される。この装置特性に基づき、パーサ 2 4 0 2 は圧縮データの適当な部分を選択し、それを通信路 2 4 0 3 へ送り、通信路 2 4 0 3 はそのデータを伸長装置 2 4 0 4 へ転送する。伸長装置 2 4 0 4 はそのデータを復号化し、復号化データをディスプレイ・モジュール 2 4 0 5 に与える。

**【 0 2 7 2 】**

本発明は、ワールド・ワイド・ウェブ、その他の画像サーバーに対する改良したサポートをデータストリームに与える。データストリームのある部分はモニタ用の低空間解像度・高画素深度の画像をサポートすることができる。別の部分は、高空間解像度・低画素深度のプリンタをサポートすることができる。データストリーム全体は非損失性伝送を提供する。これら 3 つの使い方が同じ圧縮データによりサポートされるので、ブラウザがモニタ画像、印刷画像及び非損失性画像を順番に要求するならば、余分なデータを全く送る必要がない。伝送されたモニタ画像用情報で印刷画像のために必要とされるものは、印刷画像のために再利用できる。伝送されたモニタ画像と印刷画像のための情報は、非損失画像のために再利用できる。本発明は、閲覧のための伝送時間（伝送費用）を減らし、また、サーバーに格納しなければならないデータ量を最小にする。

10

**【 0 2 7 3 】**

本発明のシステムにおいては、画像は 1 度だけ圧縮されるが、データが何であるかを示すために様々なマーカーが格納される。その後、ワールド・ワイド・ウェブ（WEB）サーバーは、表示の要求を受信し、必要な係数を提供するのであろう。WEBサーバーは、圧縮とか伸長とか何もする必要がなく、送るべきビットストリームの適当な部分を選択するだけである。

20

**【 0 2 7 4 】**

このような構文解析システムによる量子化は、可逆ウェーブレット及びコンテキストモデルによる高度の非損失性圧縮がないとしても、帯域幅の実質的な増加をもたらす。この構文解析システムは高品質の注目領域の選択にも利用できる。

**【 0 2 7 5 】**

図 3 2 は量子化選択装置を示す。一実施例では、この選択装置はソフトウェアによって、各種装置のための適切な量子化プロファイルを決定するよう構成される。画像は変換され、様々な周波数帯域のビットプレーンを捨てることによって量子化される。次に逆ウェーブレット変換が実行される。再構成画像は表示に矛盾しない何らかの方法で処理される。モニタに表示される高解像度画像については、その処理はある種のスケージングであろう。プリンタの場合には、ある種の閾値処理又はディザ処理かもしれない。同じ処理が原画像に適用され、圧縮画像と比較される。平均 2 乗誤差が例として用いられたが、どのような視覚的差異基準を用いてもよい。様々なビットプレーンの量子化による誤差を利用し、ビットレートの節減による歪みが最低となるようにビットプレーンを選択し量子化する。このプロセスは所望のビットレート又は歪みに達するまで続けられるであろう。各種画像処理操作のための代表的な量子化がいったん決まったならば、量子化をシミュレートする必要はなく、代表値が用いられるであろう。

30

**【 0 2 7 6 】**

勿論、スケージングのような単純な画像処理操作に関しては、様々な周波数帯域の量子化の効果を解析的に決定できる。その他のディザ処理やコントラスト・マスキングのような操作に関しては、シミュレーションによりほぼ最適な量子化を見つけるのはずっと簡単である。

40

**【 0 2 7 7 】**

図 3 2 において、符号ストリーム 2 5 0 0 は、量子化を含む伸長 2 5 0 1 及び非損失性伸長 2 5 0 3 を施される。その伸長結果に対し、画像処理又は歪みモデル 2 5 0 2 , 2 5 0 4 が適用される。その出力は画像であり、MSE又はHV5差モデル 2 5 0 5 のような差モデルを適用される。その差判定の結果に基づき、アラインメントが調節され（2 5 0 4）、したがって量子化も調節される。

50



## 【0278】

構文解析を容易にするため、本発明は一連のヘッダによる合図を利用する。一実施例では、本発明の符号ストリーム構造は、1つ以上のタグ値を持つ主ヘッダを含む。主ヘッダ中のタグは、符号ストリーム中のすべてのタイルのために用いられた成分の数、サブサンプリング及びアラインメント等の情報を知らせる。一実施例では、符号ストリーム中の各タイルの前に、そのヘッダがある。タイル・ヘッダの情報は、当該タイルに対してのみ適用され、また主ヘッダの情報をくつがえすかもしれない。

## 【0279】

ヘッダはそれぞれ1つ以上のタグを含む。一実施例では、インライン・マーカは無い。ヘッダ・タグは、ある既知の点からユーザがコードをリセットする所までの圧縮データの量を示す。一実施例では、どのタグもみな16の倍数のビット数である。したがって、主ヘッダ及びタイル・ヘッダはどれもみな16の倍数のビット数である。なお、どのタグも、16以外の数の倍数のビット数でも構わないことに注意されたい。どのタイルデータ・セグメントも、16の倍数のビット数になるよう適当数の0が挿入される。

## 【0280】

一実施例では、各タイル・ヘッダはそのタイル・サイズを指示するかもしれない。別の実施例では、各タイルは、どこで次のタイルが始まるか指示するかもしれない。なお、符号ストリームのバックトラックが可能なときには、そのような情報をすべて主ヘッダに挿入することにより、符号化が簡単になるかもしれない。パーサは、符号ストリームに関する情報をその量子化を実行するために利用することができる。

## 【0281】

一実施例では、タイル・ヘッダはタイルがウェーブレット方式とバイナリ方式のいずれで符号化されたか指示するかもしれない。重要性レベル・インディケータは、タイルのデータと重要性レベルとを関係付ける。重要性レベル・ロケータ (locator) は、可能な打ち切り位置を知らせる。例えば、各タイルに対し同じ歪みが望まれるときには、どの重要性レベルがその所望の歪みレベルと等しいか分かれば、パーサは符号ストリームを適切な位置で打ち切ることができる。一実施例では、各タイルは、同じビット数を持つのではなく、ほぼ同じ歪みを持つ。

## 【0282】

本発明は、重要性レベル・ロケータ・タグを持つことにより、複数のタイルと、各タイルのどこで終わるべきかの指示を持ち得るようにしている。

## 【0283】

## タグとポインタ

復号化又は構文解析に用いられる構文解析用マーカ及びその他の情報が、タグに入れてもよい。一実施例では、ヘッダは以下のルールに従うタグによって制御情報を与える。

## 【0284】

タグは固定サイズでも可変サイズでもよい。成分数、タイル数、レベル数、又はリセットもしくは所望情報の数により、タグは長さが変わってもよい。

## 【0285】

画像が構文解析され量子化されるならば、それらのタグは新しい画像特性を表すように変更される。

## 【0286】

データストリーム中のリセット点は、8の倍数のビット数になるように0を挿入される。エントロピー・コードを符号ストリームのある点でリセットできるが、その点は符号化時に決定される(しかし、それを行うことができるのは、一つの重要性レベルの符号化の終わりだけでである)。このリセットは、エントロピー・コードにおける全ての状態情報(コンテキスト及び確率)が既知の初期状態に戻されることを意味する。次に、符号ストリームは次の8の倍数のビット数まで0を挿入される。

## 【0287】

パーサは、画像を量子化する際に符号ストリーム・タグだけを手がかりとして用いる。一実施例では、この量子化処理のために、タイル長、成分長、リセット、ビット対重要性レベル、及び、重要性レベル・ロケータの各タグが用いられる。

【0288】

パーサによって画像が量子化された後、そのタグはすべて新しい符号ストリームを反映するよう変更される。これは画像と、タイルサイズ、成分数、成分のスパン、全ての長さとポインタ等々に影響を及ぼすのが普通である。さらに、画像がどのように量子化されたかを記述する情報タグも含まれる。

【0289】

表3は、本発明の一実施例における全てのタグの一覧表である。説明と用語はしばしばJPEGとは異なるが、可能な場合には同じマーカと識別子が用いられる。どの画像も少なくとも2つのヘッダ、すなわち、画像の始まりにある主ヘッダと、各タイルの始まりにあるタイルヘッダとを有する。(どの符号ストリームもみな少なくとも1つのタイルを含む)

10

3種類のタグ、すなわち区切りタグ、機能タグ及び情報タグも用いられる。区切りタグは、ヘッダ及びデータのフレーミングのために用いられる。機能タグは、利用される符号化機能を記述するために使われる。情報タグはデータに関するオプションの情報を提供する。

【0290】

【表3】

20

CREWのタグ

区切りタグ	名称	コード	主ヘッダ	タイルヘッダ
Start of image (JPEG SOI, magic number)	SOI	0xffd8	required	X
Start of CREW (CREW magic number)	SOC	0xff4f	required	X
Start of tile (similar to JPEG SOF)	SOT	0xff50	X	required
Start of scan (JPEG SOS)	SOS	0xffda	X	required
End of image (JPEG EOI)	EOI	0xffd9	required	X
機能タグ				
Image and tile size	SIZ	0xff51	required	X
Coding style	COD	0xff52	required	optional
Component alignment	ALG	0xff53	required	optional
Tile lengths, main header	TLM	0xff54	required	X
Tile lengths, tile header	TLT	0xff55	X	required
Component pointers	CPT	0xff56	X	required
Importance level reset	IRS	0xff57	X	optional
情報タグ				
Version	VER	0xff60	optional	X
Bits versus importance levels	BVI	0xff61	optional	X
Importance level locator	ILL	0xff62	X	optional
Resolution	RXY	0xff63	optional	X
Comment	CMT	0xff64	optional	optional
Quantized codestream	QCS	0xff65	X	optional

30

40

【0291】

なお、“x”は当該タグが当該ヘッダ中で用いられないことを意味する。ヘッダ中のTLMタグ又は各タイル中のTLTタグのどちらが必要とされるが、その両方は必要でない。成分ポインタが必要であるのは、2つ以上の成分があるときだけである。

【0292】

図33は本発明の符号ストリームにおける区切りタグの配置を表す。各符号ストリームは

50

、1つのSOIタグ、1つのSOCタグ、1つのEOIタグだけ（及び少なくとも1つの  
 タイル）を持つ。各タイルは1つのSOTタグと1つのSOSタグを持つ。各区切りタグ  
 は16ビットで、長さ情報を含まない。

【0293】

SOIタグは、JPEGファイルの始まりを示し、16ビットのJPEGマジックナンバ  
 ー（magic number）である。

【0294】

SOCタグはファイルの始まりを示し、SOIタグの直後にくる。SOIタグとSOCタ  
 グは全体として、ユニーク数を形成する16ビットとなる。

【0295】

SOTタグはタイルの始まりを示す。符号ストリームには少なくとも1つのタイルがある  
 。SOTはストリームがまだ同期していることを保証するためのチェックとして働く。

【0296】

SOSタグは”スキャン”の始まりを示し、その後にはタイルの実画像データが続く。SO  
 Sはタイル・ヘッダの終わりを示し、また、CREW符号ストリームには少なくとも1つ  
 のSOSがなければならない。SOSとその次のSOT又はEOI（画像の終わり）との  
 間のデータは16の倍数のビット数であり、この符号ストリームは必要であれば0が挿入  
 される。

【0297】

EOIタグは画像の終わりを示す。EOIはストリームがまだ同期していることを保証す  
 るためのチェックとして働く。符号ストリーム中に少なくとも1つのEOIがある。

【0298】

機能タグは、タイル又は画像の全体を符号化するために利用される機能を記述する。これ  
 らタグの中には、主ヘッダに用いられるが、個々のタイルの符号化中に別の値を持つ同じ  
 タグを使って覆うことができるものもある。SIZタグは画像格子の幅と高さ、タイルの  
 幅と高さ、成分数、色空間変換（必要なとき）、各成分のサイズ（画素深度）、及び成分  
 が基準格子をどのように埋めるかを定義する。このタグは主ヘッダにのみ出現し、タイル  
 ・ヘッダ中には出現しない。各タイルは、その成分のすべてに同じ特性を提供させる。こ  
 こで定義されたパラメータの多くは他のタグのためにも用いられるため、SIZタグはS  
 OCタグのすぐ後に続かねばならない。このタグの長さは、SIZの後の最初のフィール  
 ドであるLsizに保存されるが、成分数に依存する。図34はSIZタグに関する画像と  
 タイルのサイズのシンタックスを示す。

【0299】

以下は各要素のサイズと値の説明リストである。

【0300】

SIZ：マーカ。

【0301】

Lsiz：マーカを含めない、バイト数で表したタグの長さ（偶数でなければならない）  
 。

【0302】

Xsiz：画像基準格子の幅（1成分の画像又は共通のサブサンプリングによる色成分を  
 持つ画像では画像幅と同じ）。

【0303】

Ysiz：画像基準格子の高さ（1成分の画像又は共通のサブサンプリングによる色成分  
 を持つ画像では画像の高さと同じ）。

【0304】

XTsiz：1タイル画像基準格子の幅。タイルは、あらゆる成分の1標本を持てるだけ  
 の幅でなければならない。画像幅内のタイルの数は

【0305】

【外3】

10

20

30

40

50

## [Xsiz / XTsiz]

【0306】

に等しい。

【0307】

Y T s i z : 1 タイル画像基準格子の高さ。タイルは、あらゆる成分の1標本を持てるだけの高さでなければならない。画像高さ内のタイルの数は

【0308】

【外4】

10

## [Ysiz / YTsiz]

【0309】

に等しい。

【0310】

C s i z : 画像中の成分の数。

【0311】

C S s i z : 色空間変換の種類(必要なとき)。このタグは包括的ではない。(多くの多成分空間変換はここでは指定できない。それらは本発明のファイルフォーマット内でない、ほかの場所で指示される必要がある)。表4に色空間変換のための値を示す。

20

【0312】

【表4】

C S <sub>siz</sub> はパラメータの色空間変換方式

値	色空間変換
0	無し又は別途指定
1	R I C O H非損失性
2-255	予備

30

【0313】

このタグにおいて記述されるサブサンプリングは、各成分に最高解像度を利用できない画像に適用される。本発明のシステムは、最高解像度を利用できる時に重要性の低い成分のサイズを縮小する別の方法がある。

【0314】

S s i z i : 第i成分の精度(画素深度)。このパラメータ、X R s i z 及びY R s i z は全ての成分のために繰り返される。

【0315】

X R s i z i : 第i成分のX次元の大きさ。例えば、数字2は当該成分が2つの水平基準格子点に寄与することを意味する。このパラメータと、S s i z、Y R s i z は全ての成分のために繰り返される。

40

【0316】

Y R s i z i : 第i成分のY次元の大きさ。例えば、数字2は当該成分が2つの垂直基準格子点に寄与することを意味する。このパラメータと、X s i z、X R s i z は全ての成分のために繰り返される。

【0317】

r e s : 必要なときに最後に置かれる0の埋め草バイト。

【0318】

50

【表5】

画像及びタイルサイズ値

パラメータ	サイズ (ビット)	値
<b>SIZ</b>	16	0xff51
<b>Lsiz</b>	16	24 - 65534
<b>Xsiz</b>	32	1 - (232 - 1)
<b>Ysiz</b>	32	1 - (232 - 1)
<b>XTsiz</b>	32	1 - (232 - 1)
<b>YTsiz</b>	32	1 - (232 - 1)
<b>Csiz</b>	8	1 - 255
<b>CSsiz</b>	8	表4による
<b>Ssizi</b>	8	1 - 255
<b>XRsizei</b>	8	1 - 255
<b>YRsizei</b>	8	1 - 255
<b>res</b>	8	0 (必要なとき)

10

## 【0319】

CODタグは、画像又はタイルに用いられたバイナリ方式やウェーブレット方式といった符号化方式、変換フィルタ及びエントロピー・コードを記述する。このタグは主ヘッダに含まれ、またタイルヘッダにも使用できる。このタグの長さは成分数に依存する。図35は、符号化方式シンタックスを示す。表6は符号化方式のためのサイズと値を示す。

20

## 【0320】

【表6】

符号化方式値

パラメータ	サイズ (ビット)	値
<b>COD</b>	16	0xff52
<b>Lcod</b>	16	4 - 258
<b>Ccod<sup>i</sup></b>	8	表7による
<b>res</b>	8	0 (必要なとき)

30

## 【0321】

COD：マーカー。

## 【0322】

Lcod：マーカーを含めない、バイト数で表したタグの長さ（偶数でなければならない）。

40

## 【0323】

Ccod<sup>i</sup>：各成分の符号化方式。

## 【0324】

res：必要なときに最後に置かれる0の埋め草バイト。

## 【0325】

【表7】

C<sub>cod</sub> パラメータの符号化方式値

値	符号化方式
0	T S変換、F S Mコーダ
1	バイナリ、F S Mコーダ
2-255	予備

10

## 【0326】

各成分毎に、ALGタグはピラミッドレベル数と係数のアラインメントを記述する。ALGは主ヘッダに用いられ、またタイルヘッダにも用いることができる。このタグの長さは、成分数に依存し、場合によってはレベル数にも依存する。図36は、本発明の成分アラインメント・シンタックスの一例を示す。図36において、以下の成分が含まれる。

## 【0327】

ALG：このマーカーは、成分アラインメント・パラメータのサイズと値を示す。

## 【0328】

## 【表8】

20

## 成分アラインメント値

パラメータ	サイズ (ビット)	値
ALG	16	0xff53
Lalg	16	4-65535
Palgi	8	1-255
Aalgi	8	表9による
Talgi	8	表10による
Salgij	8又は16	(0-255)又は(0-65535)
res	8	0(必要なとき)

30

## 【0329】

Lalg：マーカーを含めない、バイト数で表したタグの長さ（偶数である）。

## 【0330】

Palgi：第i成分のピラミッド分解レベル数。このパラメータとAalgi、場合によってはSalgiも、各成分毎に1レコードとして繰り返される。

## 【0331】

Aslgi：第i成分のアラインメント。このテーブル・エントリ-は、係数のアラインメントを記述し、あらゆる成分のために繰り返される。表9にAalgiパラメータの値を表す。

40

## 【0332】

## 【表9】

## Aalgiパラメータのアライメント値

値	アライメント形式
0	MSEアライメント
1	厳密ピラミッド、アライメント
2	レベル2のピラミッド
3	レベル3のピラミッド
4	レベル4のピラミッド
5-253	予備
254	カスタム、16ビット
255	カスタム、8ビット

10

## 【0333】

Palg、場合によってはSalgも、各成分毎に1レコードとして繰り返される。

## 【0334】

Talgi: 表10にテール情報選択方法を示す。

## 【0335】

20

## 【表10】

## 親テール情報

値	テール情報の選択方法
0	カレント重要性レベル及びそれ以上の全ての重要性レベルに基づく
1	常に0
2	[カレント要重性レベル+画素深度+3] 重要性レベル及びそれ以上の全ての重要性レベルに基づく
3-255	予備

30

## 【0336】

Salgij: 第i成分の第jサブブロックのアラインメント値であり、当該成分のためのAalgiの値が"カスタム・アラインメント"であるときにのみ用いられる。この数は、どのカスタム・アラインメントが選ばれるかにより8ビット又は16ビットであり、また、当該成分に関し、画像のあらゆる周波数帯域のために順番に繰り返される。(バイナリ方式に関しては、Salgijは第iピラミッドレベルのアラインメント値である) Salgijが用いられる時には、Salgijと、Aalg及びPalgは各成分毎に1レコードとして繰り返される。

40

## 【0337】

res: 必要なときに最後に置かれる0の埋め草バイト。

## 【0338】

TLMタグは、画像中のあらゆるタイルの長さを記述する。各タイルの長さは、SOTタグの第1バイトから(次のタイルの)次のSOTタグの第1バイト、又はEOI(画像の終わり)までを測った長さである。言い換えれば、この長さはタイルへのポイントのリスト又はデージーチェーンである。

## 【0339】

50

符号ストリームは、単一のTLMタグ又は各タイル毎のTLTタグのいずれかを含むが、その両方は含まない。主ヘッダ中にTLMタグが使用される時には、TLTタグは用いられない。逆に、各タイルがTLTタグで終わるときには、TLMタグは用いられない。TLMヘッダ中の個々のタイル長の値は、TLMが使われなかったならば対応TLTタグのために用いられるであろう値と同じである。TLMタグの長さは、画像中のタイル数に依存する。図37はタイル長・主ヘッダのシンタックスの一例を示す。

【0340】

TLM：表11に、タイル長・主ヘッダパラメータのサイズと値を示す。

【0341】

【表11】

10

#### タイル長、主ヘッダ値

パラメータ	サイズ (ビット)	値
<b>TLM</b>	<b>16</b>	<b>0xff54</b>
<b>Ltlm</b>	<b>16</b>	<b>6 - 65534</b>
<b>Ptlmi</b>	<b>32</b>	<b>2 - (232-2)</b>

【0342】

Ltlm：マーカを含めない、バイト数で表したタグの長さ（偶数である）。

【0343】

Ptlmi：第*i*タイルのSOTマーカから次のSOT（又はEOI）マーカまでのバイト数で表した長さ。これは、画像中のあらゆるタイルのために繰り返される。

【0344】

TLTタグはカレント・タイルの長さを記述するが、この長さは、SOTタグの第1バイトから次のタイルのSOTタグの第1バイトまで（又はEOIまで）を測った長さである。言い換えれば、TLTは次のタイルへのポインタである。TLTシンタックスの一例を図38に示す。

【0345】

TLMタグかTLTタグのいずれかが必要とされ、両方は必要とされない。TLTタグは、使用される時には、全てのタイルヘッダに必要とされ、そしてTLMタグは使われない。これらのタイル長の値は両マーカとも同一である。

【0346】

TLT：表12に、タイル長・タイルヘッダパラメータのサイズと値を示す。

【0347】

【表12】

30

#### タイル長、タイルヘッダ値

パラメータ	サイズ (ビット)	値
<b>TLT</b>	<b>16</b>	<b>0xff55</b>
<b>Ltlm</b>	<b>16</b>	<b>6</b>
<b>Ptlm</b>	<b>32</b>	<b>2 - (232-2)</b>

40

【0348】

Ltlm：マーカを含めない、バイト数で表したタグの長さ（偶数である）。

【0349】

Ptlm：タイルのSOTマーカから次のSOTマーカ（又はEOIマーカ）まで

50



の長さ(バイト数)。

【0350】

CPTタグは、SOTの第1バイトより、タイル中の第1成分以外のすべての成分の第1バイトを指し示す。成分符号化データは各タイル内にノンインターリーブ形式で配置され、8ビット境界から始まる。この点でエントロピー・コードはリセットされる。

【0351】

画像が2つ以上の成分を含むときに、このタグはあらゆるタイルのタイルヘッダに使用される。この可変長タグのサイズは、画像中の成分数に依存する。成分ポインタのシンタックスの一例を図39に示す。

【0352】

CPT：表13に、成分ポインタのパラメータのサイズと値を示す。

【0353】

【表13】

成分ポインタ・タグ値

パラメータ	サイズ (ビット)	値
CPT	16	0xff56
Lcpt	16	6 - 65534
Pcpti	32	1 - (232 - 1)

【0354】

Lcpt：マーカーを含めない、バイト数で表したタグの長さ(偶数である)。

【0355】

Pcpti：カレント・タイルのSOTタグから次の成分の始まりまでのバイト数。第1成分のデータはSOSタグの直後に始まるため、Pcpt値の数は成分数より小さい。新たな成分データは8ビット境界上で始まる。

【0356】

IRSタグは、カレント・タイルのSOTタグの第1バイトよりデータ中のリセットを指し示す。これらのリセットは、符号化が完了した重要レベルの終わりの後の8ビット境界に見出される。リセットが生じる点の成分は、CPTタグ値とリセット・ポインタとの間関係によって決定できる。このタグの長さは、復号化器に利用されたリセットの数に依存する。重要性レベル・リセット・シンタックスの一例を図40に示す。

【0357】

IRS：表14に重要性レベル・リセットのパラメータのサイズと値を表す。

【0358】

【表14】

重要性レベル・リセット値

パラメータ	サイズ (ビット)	値
IRS	16	0xff57
Lirs	16	8 - 65535
Iirsi	16	1 - 65535
Pirsi	32	1 - (232 - 1)

【0359】

Lirs：マーカーを含めない、バイト数で表したタグの長さ(偶数である)。

## 【0360】

$Iirs^i$  : 第  $i$  リセットでのカレント重要性レベルの番号。この  $Iirs$  タグと、対応した  $Pirs$  タグとが一種のレコードを形成し、これは各リセット毎に繰り返される。これらのレコードは、リセットを持つ最も高い重要性レベルからリセットを持つ最低の重要性レベルへと続き、その次の成分の重要性レベルのものが続き、同様にして最後の成分まで続く順序である。

## 【0361】

$Pirs^i$  : カレント・タイルの  $SOT$  タグから第  $i$  リセットのバイトまでのバイト数。この  $Pirs$  タグと  $Iirs$  タグとが一種のレコードを形成し、これは各リセットに対し繰り返される。これらのレコードは、最小のポインタから最大のポインタへの順序でなければならない。すなわち、これらのレコードは、各リセットバイトを符号ストリーム中で出現した順に指し示す（数が小さくなるほど、物理的には先に出現するバイトを指す）。

10

## 【0362】

特定の情報タグがもっぱら情報目的のために含まれる。これらの情報タグは、復号化器のためには必要ではないが、パーサの助けとなる。

## 【0363】

例えば、 $VER$  タグはメジャー・バージョン番号及びマイナー・バージョン番号を記述する。このタグは、主ヘッダに使われる。このタグは、規定されてはいるが、画像の復号化に必要とされる機能レベルを意味しない。実は、その目的は、あらゆる復号化器及びパーサを、本発明のどのバージョンの符号ストリームも復号化及び構文解析できるようにすることである。本発明のバージョン番号のシンタックスの一例を図 4 1 に示す。

20

## 【0364】

$VER$  : 表 1 5 にバージョン番号パラメータのサイズと値を示す。

## 【0365】

## 【表 1 5】

バージョン番号値

パラメータ	サイズ (ビット)	値
$VER$	16	0xff60
$Lver$	16	4
$Vver$	8	0 - 255
$Rver$	8	0 - 255

30

## 【0366】

$Lver$  : マーカーを含めない、バイト数で表した卓の長さ（偶数である）。

## 【0367】

$Vver$  : メジャー・バージョン番号。

## 【0368】

$Rver$  : マイナー・バージョン番号。

40

## 【0369】

$BVI$  タグは、画像幅を基準にして、ビットの数を重要性レベルに関連付ける。このオプションのタグは、主ヘッダに用いられる。この可変長タグのサイズは、符号化器によって数え上げられた重要性レベルの数に依存する。ビット対重要性レベル・シンタックスの一例を図 4 2 に示す。

## 【0370】

$BVI$  : 表 1 6 に、タイル長主ヘッダ・パラメータのサイズと値を示す。

## 【0371】

## 【表 1 6】

50

## ビット対重要性レベル

パラメータ	サイズ (ビット)	値
BVI	16	0xff61
Lbvi	16	10 - 65535
Cbvii	8	1 - 255
Ibvii	16	0 - 65535
Pbvii	32	0 - (232 - 1)
res	8	0 (必要なとき)

10

## 【0372】

L b v i : マーカーを含めない、ビット数で表したタグの長さ(偶数である)。

## 【0373】

C b v i<sup>i</sup> : これは、どの成分データが記述されるのか知らせる。このC b v iパラメータはI b v i及びP b v iと共に、1レコードを形成し、これは記述されたすべての成分及び重要性レベルについて繰り返される。最初の成分の全ての重要性レベル記述、次の成分の全ての重要性レベル記述、等々と続くような順序でなければならない。

20

## 【0374】

I b v i<sup>i</sup> : カレント成分において、P b v i i内のバイト数につき符号化された重要性レベルの番号。この番号(1つ又は複数)は、レート・歪み曲線の関心点を伝えるために符号化時に選択される。このI b v iパラメータはC b v i及びP b v iとともに1レコードを形成し、これは記述されたすべての成分及び重要性レベルについて繰り返される。

## 【0375】

P b v i<sup>i</sup> : 主ヘッダとタイルヘッダ、及び、I b v i i内の重要性レベルの数に関連した全てのデータを含む符号化ファイル中のバイト数。このP b v iパラメータはC b v i及びI b v iとともに1レコードを形成し、これは記述されたすべての成分及び重要性レベルについて繰り返される。

30

## 【0376】

r e s : 必要なときに最後に置かれる0の埋め草バイト。

## 【0377】

I L Lタグは、符号化データの重要性レベルの終わりに対応した符号ストリームへのポインタを記述する。I L Lタグは、I R Sタグと似ているけれども、リセットも8ビット境界へのビット挿入もないデータを指し示す。このタグにより、パーサは、画像幅基準でほぼ同じひずみのタイルを見つけて打ち切ることが可能になる。このタグは、オプションであり、タイルヘッダ中でだけ使われる。このタグの長さは、数え上げられた重要性レベルの数に依存する。重要性レベル・ロケータのシンタックスの一例を図43に示す。

## 【0378】

I L L : マーカー。表17に、重要性レベル・ロケータのパラメータのサイズと値を示す。

40

## 【0379】

## 【表17】

## 重要性レベル・ロケータ値

パラメータ	サイズ (ビット)	値
ILL	16	0xff62
Lill	16	10 - 65535
Iilli	16	1 - 65535
Pilli	32	0 - (232 - 1)

10

## 【0380】

L i l l : マーカーを含めない、バイト数で表したタグの長さ(偶数である)。

## 【0381】

I i l l i : P i l l i 内のバイト数に関し符号化される重要性レベルの番号。それら番号はそれぞれ、レート・歪み曲線の関心点を伝達するため符号化時に選択される。この I i l l i 番号は P i l l i パラメータと共に1レコードを形成するが、これは最も速い成分において最高重要度レベルから最低重要性レベルへの順に繰り返され、以下、後の成分における重要な最高重要性レベルから最低重要性レベルまでを特定する同様レコードが続く。

## 【0382】

P i l l i : カレント・タイルの S O T の第1バイトより、当該タイルの符号化データ中の I i l l i の重要性レベルが完了するバイトを指し示す。この P i l l i 数は I i l l i パラメータと共に1レコードを形成し、これは最も速い成分において最高の重要性レベルから最低の重要性レベルへの順に繰り返され、以下、後の成分における重要な最高重要性レベルから最低重要レベルまでを特定する同様レコードが続く。

20

## 【0383】

R X Y タグは、実寸法に関する画像基準格子の X 解像度及び Y 解像度を定義する。このタグは主ヘッダにのみ用いられる。解像度(画素/単位)のシンタクスの一例を図44に示す。

## 【0384】

R X Y : 表18に、解像度(画素/単位)を指定するためのパラメータのサイズと値を示す。

30

## 【0385】

## 【表18】

## 解像度(画素/単位)値

パラメータ	サイズ (ビット)	値
RXY	16	0xff63
Lrxy	16	8
Xrxy	16	1 - 65535
Yrxy	16	1 - 65535
RXrxy	8	0 - 255
RYrxy	8	0 - 255

40

## 【0386】

L r x y : マーカーを含めない、バイト数で表したタグの長さ(偶数である)。

## 【0387】

X r x y : 単位あたりの基準格子画素数。

## 【0388】

50

$Y r x y$  : 単位あたりの基準格子ライン数。

【0389】

$R X r x y$  : X次元の単位。したがって、水平方向の解像度は、 $X r x y$  格子画素 / 10 ( $R X r x y - 128$ )メートルである。

【0390】

$R Y r x y$  : Y次元の単位。したがって、垂直方向の解像度は $Y r x y$  格子ライン / 10 ( $R Y r x y - 128$ )メートルである。

【0391】

CMTタグはヘッダ内の非構造化データを許す。このタグは、主ヘッダとタイルヘッダのいずれにも使用できる。このタグの長さは、コメントの長さに依存する。コメントのシタックスの一例を図45に示す。 10

【0392】

CMT : 表19にコメント・パラメータの大きさと値を示す。

【0393】

【表19】

### コメント値

パラメータ	サイズ (ビット)	値
CMT	16	0xff64
Lcmt	16	6-65535
Rcmt	16	表20による
Ccm ti	8	0-255
res	8	0 (必要なとき)

20

【0394】

Lcmt : マーカーを含めない、バイト数で表したタグの長さ (偶数である)。

【0395】

Rcmt : タグのレジストレーション (registration) 値。表20に、レジストレーション・パラメータの値を示す。 30

【0396】

【表20】

### Rcmtパラメータのレジストレーション値

値	レジストレーション値
0	一般用途
1-65535	レジストレーション用に予約

40

【0397】

Ccm ti : 非構造化データのバイト。任意に繰り返される。

【0398】

res : 必要なときに、最後に置かれる0の埋め草バイト。

【0399】

QCSタグは、量子化符号データがどこまで量子化済みかを記述する。パーサ又は符号化器により量子化が実行される時に、このタグは、符号化器が、重要性レベルに関しどこまで符号化すべきかを大まかに判断するのを助ける。このタグは、オプションであり、タイ 50

ルヘッダにのみ使用される。量子化符号ストリームのシンタックスの一例を図46に示す。

【0400】

QCS：表21に、量子化符号ストリームのパラメータのサイズと値を示す。

【0401】

【表21】

量子化符号ストリーム値

パラメータ	サイズ (ビット)	値
QCS	16	0xf65
Lqcs	16	6-65535
Cqcsi	8	1-255
Iqcsi	16	0-65535
res	8	0 (必要なとき)

10

【0402】

Lqcs：マーカーを含めない、バイト数で表したタグの長さ（偶数である）。

【0403】

Ci111：カレント成分の番号。このCi111番号はIqcsとともに1レコードを形成し、これは最も速い成分における最高重要性レベルから最低重要性レベルへの順に繰り返され、以下、後の成分における重要な最高重要性レベルから最低重要レベルまでを特定する同様レコードが続く。

20

【0404】

Iqcsi：これは、符号化データの少なくとも一部分が残っている重要性レベルである。当該点から次のリセットまでに残っているデータは全て、打ち切り済み（量子化済み）である。

【0405】

res：必要に応じて最後に置かれる0の埋め草バイト。

30

【0406】

#### 損失性係数再構成

本発明は、一実施例において、値を所定の整数値の集合に丸めることで損失性再構成を行う。例えば、0と31の間の全ての係数は0に量子化され、32～63の間の全ての係数は32に量子化される等々である。図47は、量子化しないときの係数の代表的分布を示す。各係数の最も下のビットが分かっている場合に、そのような量子化が行われるかもしれない。別の実施例では、各値域の中央の値が、その係数群を表すより正確な値を提供するかもしれない。例えば、64と127の間の全ての係数が95に量子化される。値がある点へ量子化されるとき、その点は再構成点と呼ばれる。

【0407】

画像間の差異により、得られる分布は形がゆがむ。例えば、図47中の曲線2701と曲線2702を比較されたい。

40

【0408】

本発明においては、再構成点は、その分布に基づいて選ばれる。一実施例では、分布が推定され、その推定分布に基づき再構成点を選ばれる。推定分布は、既に分かったデータに基づき生成される。データを収集する以前は、デフォルトの再構成点が用いられるであろう。このように、本発明は、適応的な損失性再構成方法を提供する。さらに、本発明は、係数再構成を改善する非反復の方法である。分布の差異によって値域の使用が不均一になることを補償するため、本発明は次のように規定する。

【0409】

50

【数 1 6】

 $s^2 = \text{sample variance}$  $Q = \text{Quantization}$  $\sigma^2 = \frac{2}{\alpha^2} = \text{True variance}$ 

$$\alpha = -\frac{1}{Q} \ln \left[ \frac{Q^2 + 2s^2 - \sqrt{Q^4 + 8Q^2s^2}}{2(s^2 - Q^2)} \right]$$

10

【0 4 1 0】

ただし、 $2^s$  は利用できるデータを基に復号化器により測定された標本分散であり、 $Q$  は復号化器に知らされた量子化である。次に、非ゼロ係数を 0 から遠ざけることによって、それを修正する。

【0 4 1 1】

【数 1 7】

$$iQ \rightarrow iQ + \left[ \frac{1}{\alpha} - \frac{Q}{e^{\alpha Q} - 1} \right] \quad i > 0$$

$$iQ \rightarrow +iQ - \left[ \frac{1}{\alpha} - \frac{Q}{e^{\alpha Q} - 1} \right] \quad i < 0$$

20

【0 4 1 2】

ただし、 $i$  は任意の整数である。

【0 4 1 3】

一実施例では、全部の復号化が完了した後に、非ゼロ係数はすべて、ある再構成レベルに調整される。この調整をするためには、各係数を読み込み、恐らく修正し、そして書き込むことが必要である。

30

【0 4 1 4】

別の実施例では、各係数の各ビットプレーンが処理される時に、その係数が非ゼロならば、その係数の適当な再構成値が記憶される。復号化が止まった時に、全係数がそれらの適当な再構成値に設定される。こうすることにより、再構成レベルの設定のため別にメモリを経由する必要がなくなる。

【0 4 1 5】

カラー

本発明は、カラー画像（及びデータ）に適用できる。図 1 の多成分処理機構 1 0 1 は、カラーデータのために必要とされる処理を実行する。例えば、Y U V 色空間には、3 つの成分、つまり Y 成分、U 成分、V 成分があり、各成分は別々に符号化される。

40

【0 4 1 6】

一実施例では、各成分のエントロピー符号化データは、他の成分のエントロピー符号化データから分離される。この実施例においては、成分のインターリーブはない。成分別にデータを分けることは、ピラミッド・アラインメントと組み合わせられると、復号化器又はパーサが異なった成分を容易に別々に量子化できるようにするのに役立つ。

【0 4 1 7】

他の実施例では、異なった成分のエントロピー符号化データが周波数帯域単位又は重要性レベル単位でインターリーブされる。これは、M S E アラインメントと組み合わせられると、共通の打ち切りを全成分のデータの量子化に利用できるのので有益である。このインター

50

リーピング方式のためには、符号化器が異なった成分の周波数帯域間又は重要性レベル間の関係を提供する必要がある。周波数帯域又は重要性レベルはかなり大量の符号化データであろうから、パーサ又は復号化器はマーカーを利用し成分を独立に量子化できるであろう。

**【0418】**

さらに別の実施例では、異なった成分のエントロピー符号化データは、画素毎又は係数毎にインターリーブされる。これは、MSEアラインメントと組み合わせられると、全成分に共通の打ち切りが作用するので有益である。画素単位のインターリーブの場合、復号化器及びパーサは符号化器で定義されのとおり成分間関係を利用しなければならない。

**【0419】**

本発明によれば、同じシステムでサブサンプリングを実行できる。

**【0420】**

一実施例では、各成分は別々に記憶される。伸長装置及びパーサを使うことにより、損失性出力画像を生成する時には、別々の成分メモリのそれぞれから分解レベル及び成分の選択されたものだけが取得されるであろう。例えば、YUV色空間において、Y色成分については分解レベルの全部が取得されるであろうが、U成分とY成分については第1分解レベル以外の分解レベルがすべて取得されるであろう。結果として得られる画像の組合せは、4:1:1画像である。なお、メモリに格納されているデータの異なった部分を用いることにより、別の型式の画像を得ることもできる。

**【0421】**

多くの型式の多成分画像を処理可能である。画像データは、YUVのほかに、RGB（赤、緑、青）、CMY（シアン、マゼンタ、黄）、CMYK（シアン、マゼンタ、黄、黒）又はCCIR601 YCrCbでもよい。多重スペクトル画像データ（例えば、リモートセンシング・データ）も用い得る。RGBやCMYのような視覚的データに対しては、米国特許出願第08/436,662号（1995年5月8日受理、"Method and Apparatus for Reversible Color Compression"に述べられているような非損失性色空間変換を利用できる。

**【0422】****ビット抽出**

本発明は、ビット抽出を高めるようにコンテキストモデルを計算しビットを符号化することができる。具体的には、ヘッドビットのためのコンテキストモデルは、隣接画素より与えられる情報を基礎にしている。しばしば、特に損失性圧縮を行う時に、このコンテキストは0である。ヘッドビット・コンテキストの近似統計量のため、本発明はヘッドビットのためのコンテキストを保持する機構を提供する。

**【0423】**

一実施例では、符号化に先だってメモリがクリアされる。コンテキストは、その親、隣接画素の一つ、又は注目画素が変わるまで、そのままである。変化した時に、影響を受ける全てのコンテキストに関しコンテキスト・メモリが更新される。テール情報を利用するときには、隣接画素と子だけが更新される。ヘッドビットがオンの時に1係数につき1度だけメモリが更新される。

**【0424】**

一実施例では、各係数は、符号(sign)の1ビット、テールオン情報の4ビット、コンテキストの8ビット、その後続く係数の19ビットからなる32ビット整数として記憶される。係数の一例を図48に示す。

**【0425】**

一実施例では、テールオン情報の4ビットを利用して5つの異なったケースを生成する。

**【0426】**

テールオン情報の4ビットの値が0のケースにおいては、カレント係数の絶対値ビットのカレント・ビットプレーンのビットは、コンテキストビットを利用して符号化される。該ビットが0ならば、プロセスは終了する。該ビットが1ならば、係数の符号が符号化され

10

20

30

40

50



る。それから、テールオン情報の第1ビットが反転され、北、北東、西、南、東及び4つの子のコンテキストが更新され、プロセスは終了する。

【0427】

テールオン情報の4ビットの値が1のケースにおいては、カレント係数の絶対値ビットのカレント・ビットプレーンのビットは当該ケースのための一定のコンテキストを使って符号化される。テールオン情報の第2ビットが反転される。カレント係数の東と子のコンテキストが更新される。プロセスは終了する。

【0428】

テールオン情報の4ビットの値が7のケースにおいては、カレント係数の絶対値ビットのカレント・ビットプレーンのビットは当該ケースのための一定のコンテキストを使って符号化される。テールオン情報の第3ビットが反転される。どのコンテキストも更新不要である。プロセスは終了する。

10

【0429】

テールオン情報の4ビットの値が3のケースにおいては、カレント係数の絶対値ビットのカレント・ビットプレーンのビットは、当該ケース用の一定のコンテキストを用いて符号化される。テールオン情報の第4ビットが反転される。カレント係数の東と子のコンテキストが更新される。プロセスは終了する。

【0430】

テールオン情報の4ビットの値が15のケースにおいては、カレント係数の絶対値ビットのカレント・ビットプレーンのビットは、当該ケースのための一定のコンテキストを使って符号化される。テールオン情報のどのビットも反転不要である。プロセスは終了する。

20

【0431】

図48は本発明の係数の例を示す。図48において、係数2801は、符号ビット2802と、それに続くテールネオン情報ビット2803、それに続くコンテキストビット2804、それに続く係数絶対値ビット2805とからなる。前述のプロセスが図49のフローチャートに示されている。

【0432】

変化が生じた時に全コンテキストを更新する当該手法を使うことにより、ヘッドビットが圧倒的に0である限り、コンテキスト・モデリングが高速に働く。特に損失性符号化の場合にそうである。

30

【0433】

可逆ウェーブレット係数のハフマン符号化

本発明は、一実施例において、ハフマン符号化を使ってウェーブレット係数を符号化する。ハフマン符号化のためのアルファベットは2つの部分からなる。第1の部分は0係数のランの長さに等しく、第2の部分は0でないターミネータ(terminator)係数のハッシュ値である。図53にアルファベット・フォーマットを示すが、これは0係数の数、換言すれば、そのランの長さを示す4ビットと、それに続く0から15までのハッシュ値を表す4ビットとからなる。

【0434】

このハッシュ値は値Nであり、このNは0でないターミネータ係数の絶対値の、2を底とする対数の整数部分である。一実施例では、このハッシュ値は値Nを表すのに必要なビット数である。例えば、 $N = -1$ 、1の場合、ハッシュ値は1である。他方、 $N = -3$ 、 $-2$ 、2、3の場合、値Nを表すのに必要なビット数は2である。同様の対応はJPEGに用いられている。

40

【0435】

このようなシチュエーションでは、許容される0係数のランの最大長は15である。ランが15を超えるときには、0が16個のランの後に新たなランが続くことを表すため特殊なトークンが使われるであろう。このような例外トークンの一つは、最初の4ビットと最後の4ビットの両方とも全部0である。一実施例では、2番目の4ビットが0の16個のトークンが全部、例外ケースのために用いられる。したがって、256個の8ビットのハ

50

フマン・トークンがある。

【0436】

一実施例では、ハフマン・トークンに関するテーブルが作られる。一実施例では、そのテーブルが全ての画像に対して用いられる。別の実施例では、多くのテーブルが作成され、量子化に応じて1つの特定のテーブルが選ばれる。各テーブルは、量子化しようとするビット数に基づいて選択される。すなわち、量子化するビット数が1ビット、2ビット、3ビット等々であるかによって、テーブルがそれぞれ選択されるわけである。別の実施例では、ハフマン符号は特定画像向けのものであり、画像と一緒に記憶/伝送される。

【0437】

テーブルを利用するために、一つのハフマン・トークンが生成される。そして、このトークンが、それが符号化されるテーブルに送られる。

10

【0438】

ハフマン・トークンは0のランの長さ及び非0のターミネータ・シンボルのハッシュ値を特定するが、ターミネータ・シンボルを一意的に特定するために割増のビットが必要になる。本発明の一実施例は、これら割増ビットを用意する。ハフマン・トークンが(例えばテーブル等から得られる)ハフマン符号語で置き換えられた後、ターミネータ・シンボルのハッシュ値に等しい割増ビットが書かれる。例えば、-1, 1のケースでは割増の1ビットが書かれるが、-3, -2, 2, 3のケースにおいては割増の2ビットが書かれる。このように、本発明は、ターミネータ・シンボルを一意的に特定する、割増ビットによってサイズが可変のハフマン符号化を提供する。

20

【0439】

なお、他のm元コードを用いてもよい。例えば、あるアルファベットとm元符号を0係数のために用い、別のアルファベットとm元符号をハッシュ値のために用いてもよい。

【0440】

一実施例では、量子化レベル毎のハフマン・テーブルのセットが予め計算され、殆どの画像に対して利用される。様々なテーブル間で選択するために、あるテーブルを使用中に圧縮がモニタされるであろう。そのテーブルを使用した結果に基づいて、スキューがもっと大きい又は小さいテーブルへの切り替えが行われるであろう。

【0441】

本発明の係数はすべて、あるバッファに入れられる。各バッファ毎に、どのテーブルを使用すべきかの決定がなされるであろう。8つのハフマン・テーブルのどれを利用すべきか指示するため、3ビットと1つのヘッダが用いられるかもしれない。しかして、そのヘッダを知らせることによって、テーブル選択がなされるであろう。

30

【0442】

係数が符号化される順序は重要である。従来技術の係数符号化では、例えばJPEGでは、係数はジグザグ順に圧縮されることに注意されたい。本発明においては、係数全部があるバッファ内にあるので、ジグザグ順にすることはできない。ジグザグ順は、低い周波数から高い周波数への順序と理解されるなら、埋め込みウェーブレットによる圧縮(ツリー順)に拡張することができる。

【0443】

一実施例では、バッファ全体について直線的な順序で係数が符号化される。そのような例を図54に示す。なお、この実施例において、平滑係数の最初のブロックは除外されることに注意されたい。

40

【0444】

別の実施例では、すべてのブロックは、低い周波数のブロックより高い周波数のブロックへと、ラスタ順に符号化される。そのような例を図54(B)に示す。メモリの制約のため、1つの周波数パスの全部は、別の周波数パスが始まる前に完了しないかもしれない。メモリによって制限される場合、もう一つの方法は1つのツリーを一度に符号化する方法である。ルートから初めて、すべてのツリーが横方向に符号化される。ただし、平滑係数であるところのルートは含めない。この方法が図54(C)に示されている。図54(C)

50

C)には最初のツリーが示されており、最初のサブブロックのセットより1ラインが取られ、その次のサブブロックのセットより2ラインが取られ、その次のサブブロックのセットより4ラインが取られる。これらラインは他のラインが利用可能になる以前に利用可能であるため、このような実施例が可能である。

【0445】

残りのツリーが0係数からなることを示すため、例外トークンを保存してもよい。これは、16個の0を示す同じトークンが何度も何度も使用されないようにする。

【0446】

一実施例では、全ての重要性レベルがハフマン符号化によって符号化される。別の実施例では、複数の重要性レベルからなる1又は複数のグループがハフマン符号化によって符号化される。別々のグループ毎に全ての重要性レベルをハフマン符号化により符号化してもよいし、あるいは、一部の重要性レベルをハフマン符号化で符号化し、残りの重要性レベルを水平コンテキストモデルとバイナリ・エントロピー・コードにより符号化してもよい。

10

【0447】

重要性レベルの1グループのハフマン符号化による符号化は、以下のように行われる。そのグループ内の重要性レベルの係数のビットが全てヘッドビットのときには、その係数は0係数として(多分、ラン・カウンタの一部として)ハフマン符号化される。その係数のビットが全てテールビットならば、それらビットは(多分、ランを終結させる)割増ビットとして符号化される。ハフマン符号語は使われない。その係数のビットが(ヘッドビット又はテールビットのほかに)に符号(sign)ビットを含んでいるときには、(多分、ランを終結させる)ハフマン符号語と割増ビットの両方が符号化される。

20

【0448】

複数の重要性レベルをハフマン符号化すれば、実行コストは減少する。しかし、ハフマン符号化データの中途での打ち切りは、レート・歪みの悪化を招く。重要性レベルのグループをハフマン符号化すれば、レート・歪みが良好になるようグループの始まり/終わりの打ち切りが可能になる。用途によっては、限定数の必要とされる量子化点が符号化時に分かっている。量子化点のない重要性レベルは、それに続くレベルと一緒にしてハフマン符号化することができる。

【0449】

用途

本発明は多くの用途に利用できる。そのような用途のいくつかを例として以下に述べる。具体的には、解像度が高く画素深度が大きいハイレンドの用途及びアーティファクト(artifact)を許容しない用途に、本発明を利用できる。本発明によれば、ハイレンドの用途は高品質環境で最高品質を維持でき、同時に、帯域幅、データ記憶又は表示機能がさらに制限される用途でも同じ圧縮データを利用可能である。これはまさに、ウェブ・ブラウザのような近頃の画像応用分野に一般に要求される装置独立な表現である。

30

【0450】

画素深度の深い画像(10ビット~16ビット/画素)に対する本発明の優れた非損失圧縮性能は、医用画像のために理想的である。非損失性圧縮のみならず、本発明は、ブロックベース圧縮装置に知られている多くのアーティファクトのない真の損失性圧縮装置である。本発明を利用することに由来する損失性アーティファクトは、急峻なエッジに沿う傾向があるので、人間の視覚系の視覚マスキング現象によって見えないことが多い。

40

【0451】

本発明は、画像が非常に高解像度で高い画素深度を持つことの多いプリプレス(pre-press)業に関連した用途に利用できる。本発明のピラミッド分解によれば、プリプレス・オペレータが(モニタ上の)画像の低解像度損失性バージョンに対し画像処理操作を行うのが容易である。操作が終わったならば、同じ操作を非損失性バージョンに対して実行できる。

【0452】

50

本発明は、圧縮しないと送信に要する時間があまりに長くなりやすいファクシミリ文書の用途にも適用可能である。本発明によれば、様々な空間解像度及び画素解像度のファクス装置より、非常に高品位の画像出力が可能になる。

【0453】

本発明は、圧縮を必要とする画像アーカイブシステムに、特に記憶容量を増加させるために、利用することもできる。本発明の装置独立な出力は、帯域幅が異なる資源、メモリ及びディスプレイを持つシステムにより画像アーカイブシステムをアクセスでき、有益である。本発明のプログレッシブ伝送機能は、ブラウジングのためにも有益である。最後に、画像アーカイブシステムの出力装置用に望ましい非損失性圧縮が本発明により提供される。

10

【0454】

本発明の非損失性又は高品質損失性データストリームの階層プログレッシブ性により、本発明はワールド・ワイド・ウェブ用に、特に装置独立性、プログレッシブ伝送及び高品質が必須な場合に理想的である。

【0455】

本発明は、衛星画像、特に高画素深度及び高解像度になる傾向のある衛星画像にも適用できる。さらに、衛星画像の用途は通信路の帯域幅が制限される。本発明はフレキシビリティがあり、またプログレッシブ伝送特性があるので、本発明を利用すれば人間による画像のブラウジング又はプレビューが可能になる。

【0456】

ATMネットワークのような”固定レート”で帯域幅が制限される用途は、データが利用可能な帯域幅をオーバーフローしたときにデータを減少させる手段を必要とする。しかしながら、十分な帯域幅があるときには（あるいはデータが高度に圧縮可能なときには）、品質上の不利益があってはならない。同様に、コンピュータや他の画像装置におけるメモリが制限されたフレーム記憶装置のような”固定サイズ”の用途も、メモリが満杯になったときにデータを減少させる手段を必要とする。繰り返すが、適当なメモリ量に非損失圧縮することが可能な画像に対して不利益があってはならない。

20

【0457】

本発明の埋め込み符号ストリームは、これら両方の用途にかなう。埋め込み操作は、損失性画像の伝送又は記憶のために符号ストリームが切り捨てもしくは打ち切りされることを無条件に許す。切りつめもしくは打ち切りが必要でなければ、画像は非損失で届く。

30

【0458】

要するに、本発明は、単一連続階調画像圧縮システムを提供する。本発明のシステムは、同じ符号ストリームに対して非損失性かつ損失性であり、埋め込みの量子化（符号ストリームに含まれる）を利用する。本発明のシステムはまた、ピラミッド型であり、プログレッシブであり、補間手段を提供し、かつ、実施が容易である。したがって、本発明はフレキシブルな”装置独立の”圧縮システムを提供する。

【0459】

統合型の損失性及び非損失性圧縮システムは非常に有用である。同じシステムで最新の損失性及び非損失性圧縮を実行でき、その上、同じ符号ストリームである。このシステムは、画像の非損失性符号を保存するか打ち切って損失性バージョンにするかを、符号化中、符号ストリームの格納又は伝送中あるいは復号化中に決定することができる。

40

【0460】

本発明により提供される損失性圧縮は、埋め込み量子化によって達成される。すなわち、符号ストリームは量子化を含んでいる。実際の量子化（又は視覚的重要性）レベルは、復号化器又は通信路との相関点要素であることもあり、必ずしも符号化器との相関的要素ではない。バンド幅、記憶及びディスプレイ資源が許すなら、画像は非損失で復元される。そうでないならば、画像は最も制約された資源に要求されるだけ量子化される。

【0461】

本発明に用いられるウェーブレットはピラミッド型であり、差分画像のない、画像の1 /

50

2分解が実行される。これは非常に特殊な階層分解である。画像のブラウジングのため又は低解像度装置による表示のために縮小画像(thumbnails)を必要とする用途に、本発明のピラミッド性は理想的である。

【0462】

本発明における埋め込みの使い方はプログレッシブであり、より具体的にはビットプレーン順である、すなわちMSBの後に下位ビットが続く順である。具体的には本発明はウェーブレット領域においてプログレッシブであるが、空間領域及びウェーブレット領域の両方ともプログレッシブに分解してもよい。プリンタのような、空間解像度はあるが画素解像度は低い用途にとって、本発明におけるビットのプログレッシブな順序づけは理想的である。これらの特徴を同一符号ストリームで得られる。

10

【0463】

本発明は、ソフトウェアでもハードウェアでも比較的容易に実施できる。ウェーブレット変換は、ハイパス、ローパスの各係数ペアにつき4つの加算/減算操作と、いくつかのシフトだけで計算することができる。埋め込み及び符号化は、単純な"コンテキストモデル"とバイナリ又はm元"エントロピー・コード"によって実行される。このエントロピー・コードは、有限状態マシン、並列コード又はハフマン・コードによって実現できる。

【0464】

【発明の効果】

以上の説明から明らかな如く、本発明のデータ圧縮システムによれば、良好なエネルギー集中をもたらす変換を利用し画像データ等の効率的な損失性又は非損失性の圧縮が可能であり、また、自然連続階調の画像、2値/ノイズフリー/浅画素深度の画像、及び、その両方の種類のデータを含む画像を同じシステムで適切に処理することができ、さらに、様々な解像度/品質の画像フォーマットを柔軟にサポートすることができる。

20

【図面の簡単な説明】

【図1】本発明の圧縮システムの一実施例のブロック図である。

【図2】バイナリ方式における各ビットプレーンの各ビットに対するコンテキストモデルの可能な幾何学的関係の一例を示す図である。

【図3】バイナリ方式における各ビットプレーンの各ビットに対するコンテキストモデルの可能な幾何学的関係の一例を示す図である。

【図4】第1レベルの分解を示す図である。

30

【図5】第2レベルの分解を示す図である。

【図6】第3レベルの分解を示す図である。

【図7】第4レベルの分解を示す図である。

【図8】前後2レベル間の親子関係を示す。

【図9】TT変換だけを利用するウェーブレット分解過程の一例を示す図である。

【図10】TT変換とS変換を利用するウェーブレット分解過程の一例を示す図である。

【図11】画像のタイリングの説明図である。

【図12】ビット・シグニフィカンス表現の例を示す図である。

【図13】本発明における係数サイズを示す図である。

【図14】本発明における係数アラインメントのために使われる周波数帯域用乗数の例を示す図である。

40

【図15】符号ストリームの構成の一例を示す図である。

【図16】係数(又は画素)間の隣接関係を示す図である。

【図17】テール・ビット処理プロセスのフローチャートである。

【図18】本発明の符号化プロセスの一例のフローチャートである。

【図19】本発明の復号化プロセスの一例のフローチャートである。

【図20】本発明のモデリング・プロセスのフローチャートである。

【図21】モデリング・プロセスに利用可能なテンプレートを示す図である。

【図22】TT変換フィルタの一部分の一例を示すブロック図である。

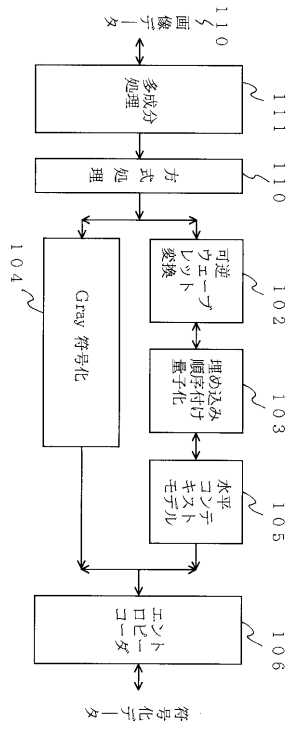
【図23】本発明のスクロール・バッファの説明図である。

50

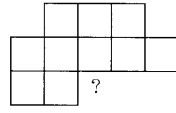
- 【図24】本発明に採用されるメモリ操作の説明図である。
- 【図25】3レベル用メモリ・バッファの2次元表現を示す図である。
- 【図26】本発明の符号ストリームの一例を示す図である。
- 【図27】パーサを備えた圧縮システムのブロック図である
- 【図28】図27の圧縮システムに対応する伸長システムのブロック図である。
- 【図29】コンテキスト従属関係を示す図である。
- 【図30】画素深度及び空間解像度の面から定義された用途を示す図である。
- 【図31】パーサ、復号化器及びそれらの出力装置との相互作用の一例を示すブロック図である。
- 【図32】量子化選択装置の一例を示すブロック図である。 10
- 【図33】符号ストリーム中の区切りタグの配置を示す図である。
- 【図34】S I Zタグの説明図である。
- 【図35】C O Dタグの説明図である。
- 【図36】A L Gタグの説明図である。
- 【図37】T L Mタグの説明図である。
- 【図38】T L Tタグの説明図である。
- 【図39】C P Tタグの説明図である。
- 【図40】I R Sタグの説明図である。
- 【図41】V E Rタグの説明図である。
- 【図42】B V Iタグの説明図である。 20
- 【図43】I L Lタグの説明図である。
- 【図44】R X Yタグの説明図である。
- 【図45】C M Tタグの説明図である。
- 【図46】Q C Sタグの説明図である。
- 【図47】損失性再構成のための典型的分布を示すグラフである。
- 【図48】典型的な係数を示す図である。
- 【図49】テーブル情報解析プロセスのフローチャートである。
- 【図50】M S Eアラインメント法を説明するための図である。
- 【図51】ピラミッド・アラインメント法を説明するための図である。
- 【図52】メモリ記憶係数とアラインメントの間の典型的な関係を示す図である。 30
- 【図53】符号語の一例を示す図である。
- 【図54】ハフマン符号化法による係数の構文解析の方法を説明するための図である。
- 【図55】ユニットバッファを用い第2レベルのウェーブレット分解を実行する場合の2Dメモリの中間形式を示す図である。
- 【図56】ユニットバッファを用い第3レベルのウェーブレット分解を実行する場合の2Dメモリの中間形式を示す図である。
- 【符号の説明】
- 101 入力画像データ
- 102 可逆ウェーブレット変換ブロック
- 103 埋め込み順序付け量子化ブロック 40
- 104 グレイ(G r a y)符号化ブロック
- 105 水平コンテキストモデル・ブロック
- 106 エントロピー・コーダ
- 110 方式選択機構
- 111 多成分処理機構
- 1001 ヘッダ
- 1002 符号化単位
- 1003 L L係数
- 1004 第1ビットプレーン
- 1005 第2ビットプレーン 50

1 0 0 6	最終ビットプレーン	
1 5 0 1	乗算器	
1 5 0 2	加算器	
1 5 0 3	乗算器	
1 5 0 4	乗算器	
1 5 0 5	加算器	
1 6 0 1	ラインアクセスバッファ	
1 6 0 2	バッファ	
1 9 0 1	ヘッダ	
1 9 0 2	LL係数	10
1 9 0 3	エントロピー符号化データ	
2 1 0 1	圧縮されていない原画像	
2 1 0 2	圧縮装置	
2 1 0 3	マーカ付き非損失性圧縮ビットストリーム	
2 1 0 4	パーサ	
2 1 0 6	通信路又は記憶装置	
2 1 0 7	伸長装置	
2 1 0 8	伸長画像	
2 4 0 1	マーカ付きの非損失性圧縮データ	
2 4 0 2	パーサ	20
2 4 0 3	通信路	
2 4 0 4	伸長装置	
2 4 0 5	ディスプレイ・モジュール	
2 5 0 0	符号ストリーム	
2 5 0 1	量子化を含む伸長	
2 5 0 2	画像処理又は歪みモデル	
2 5 0 3	非損失性伸長	
2 5 0 4	画像処理又は歪みモデル	
2 5 0 5	MSE又はHVS差モデル	
2 5 0 6	アラインメント調整	30
2 8 0 1	係数	
2 8 0 2	符号ビット	
2 8 0 3	テーブルオン情報	
2 8 0 4	コンテキストビット	
2 8 0 5	係数ビット	

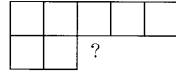
【 図 1 】



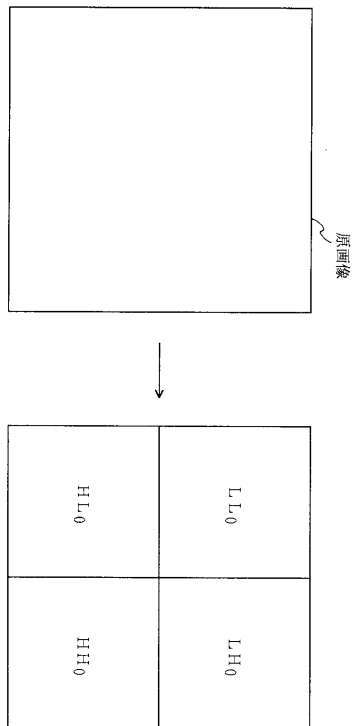
【 図 2 】



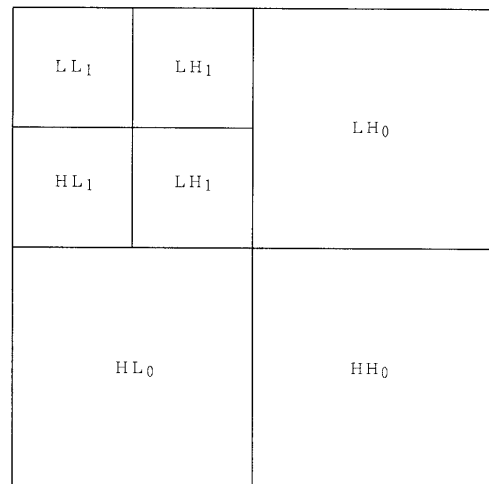
【 図 3 】



【 図 4 】

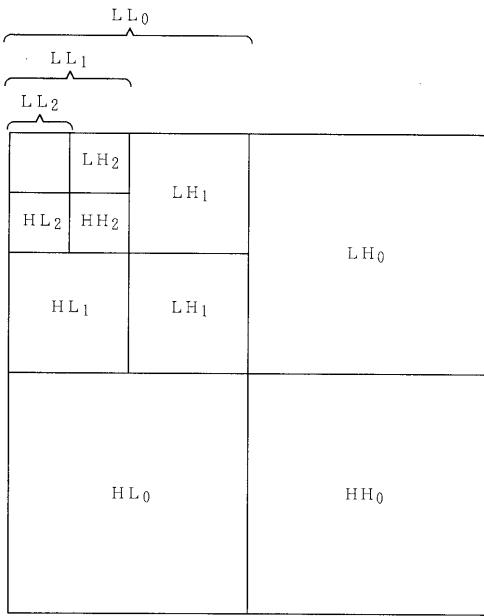


【 図 5 】

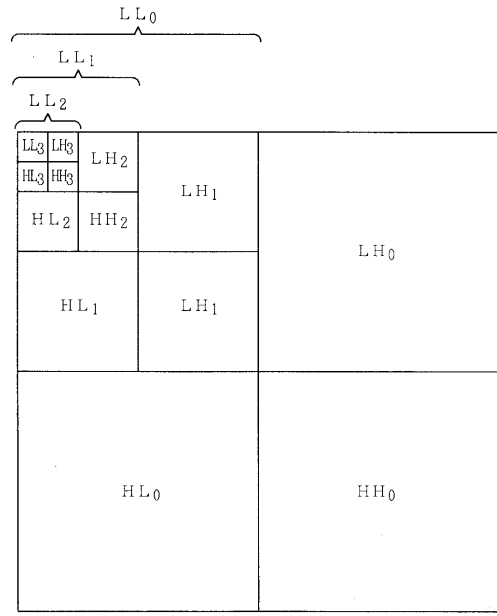




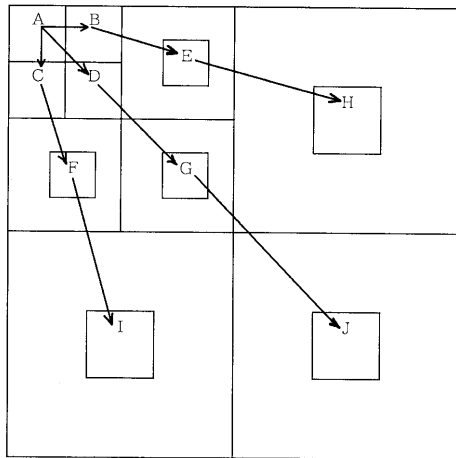
【 図 6 】



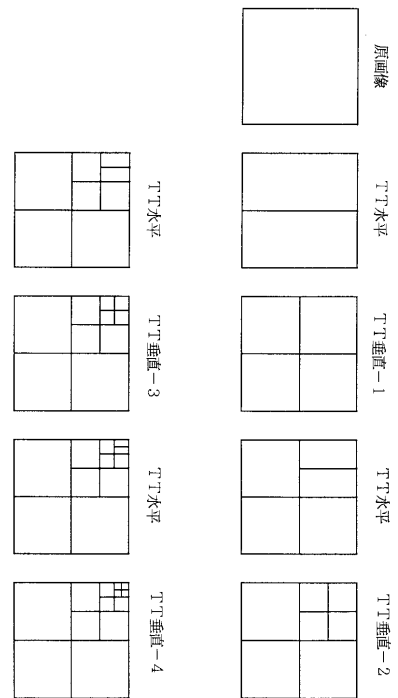
【 図 7 】



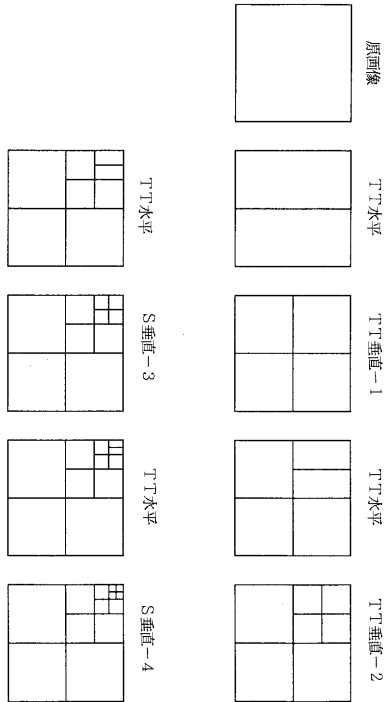
【 図 8 】



【 図 9 】



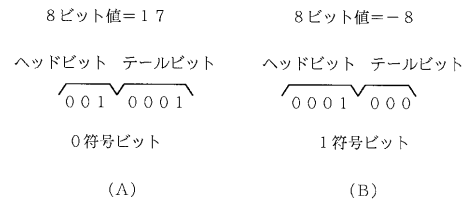
【 図 1 0 】



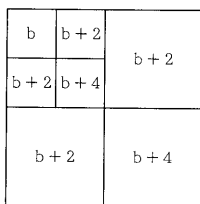
【 図 1 1 】

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35
36	37	38	39	40	41

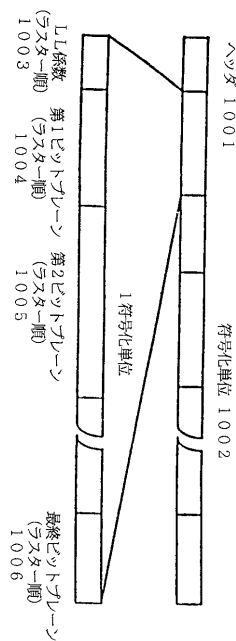
【 図 1 2 】



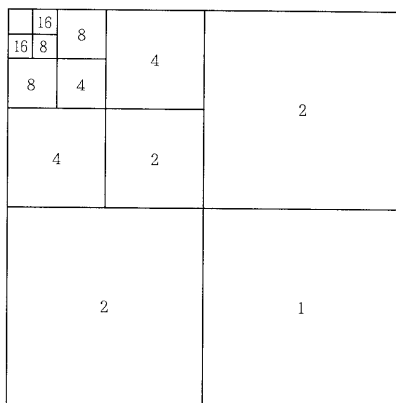
【 図 1 3 】



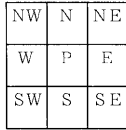
【 図 1 5 】



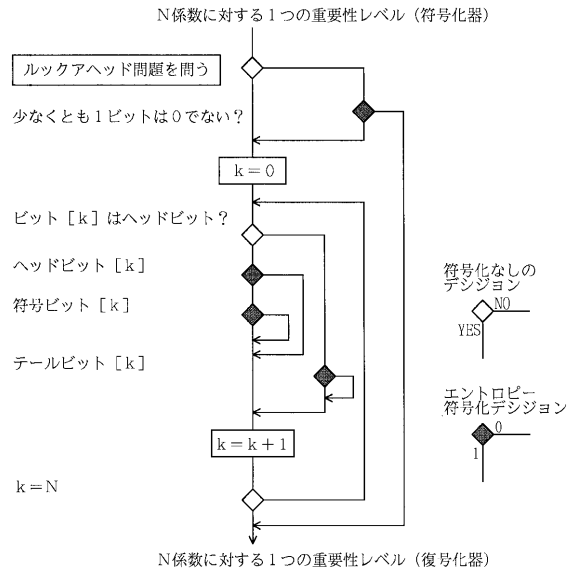
【 図 1 4 】



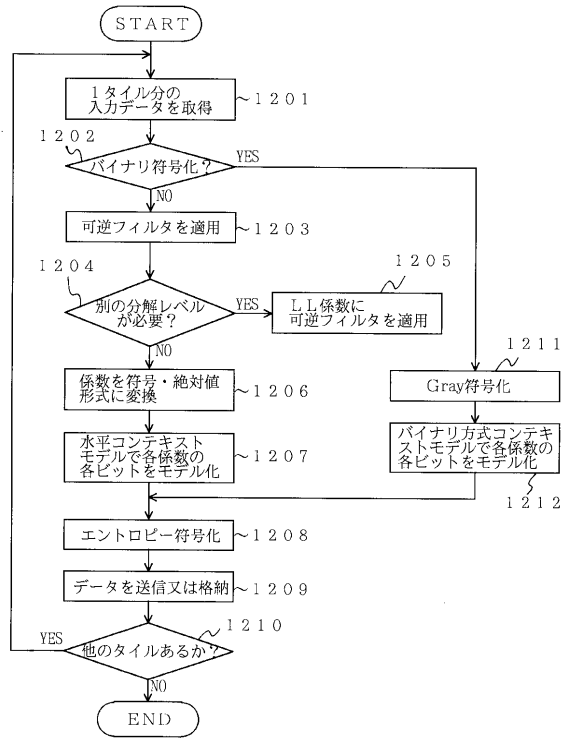
【図16】



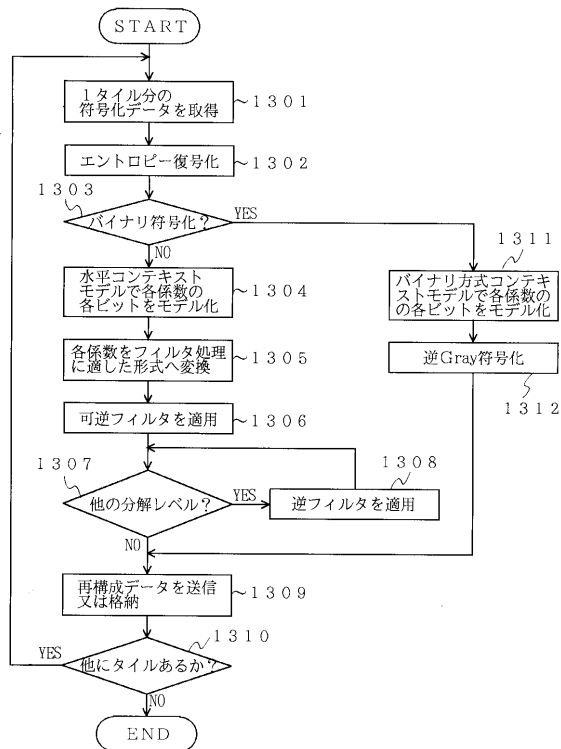
【図17】



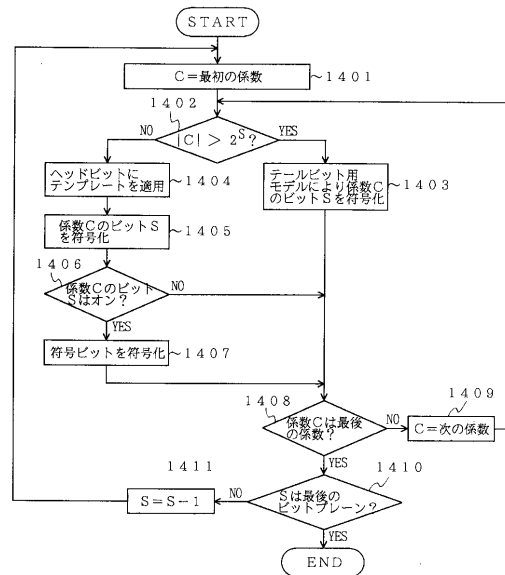
【図18】



【図19】

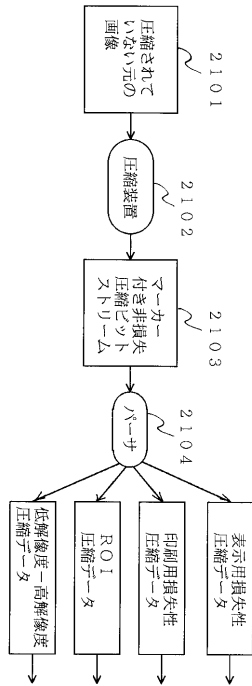


【図20】

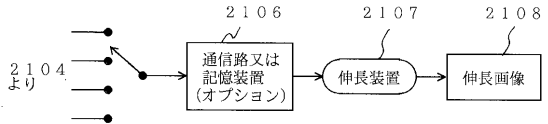




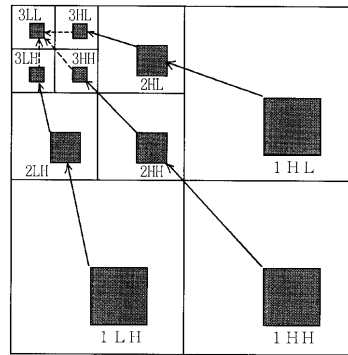
【図 27】



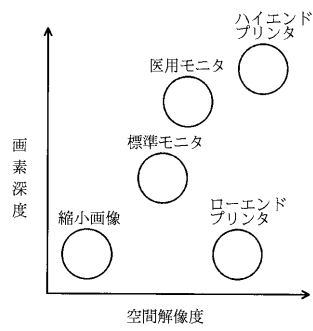
【図 28】



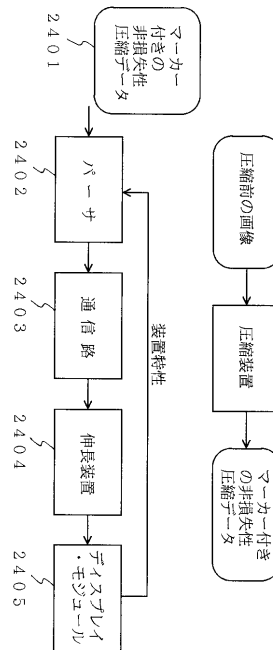
【図 29】



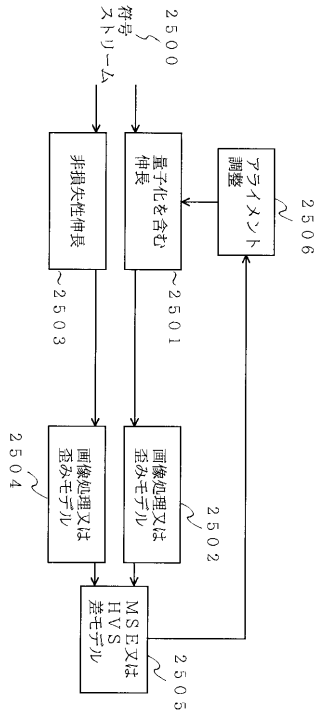
【図 30】



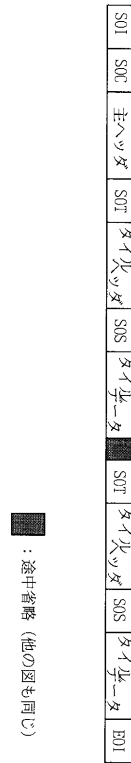
【図 31】



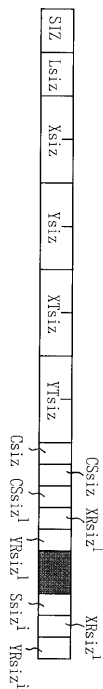
【 図 3 2 】



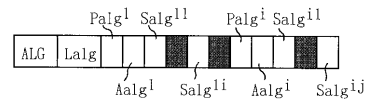
【 図 3 3 】



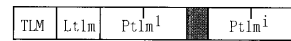
【 図 3 4 】



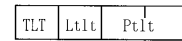
【 図 3 6 】



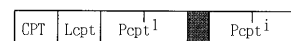
【 図 3 7 】



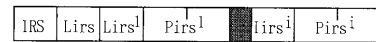
【 図 3 8 】



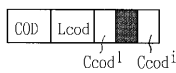
【 図 3 9 】



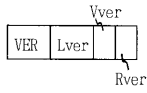
【 図 4 0 】



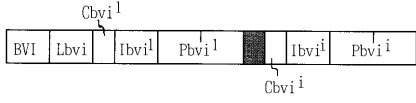
【 図 3 5 】



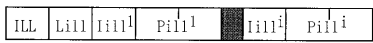
【 4 1 】



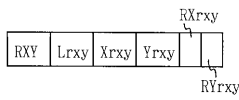
【 4 2 】



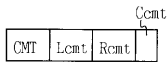
【 4 3 】



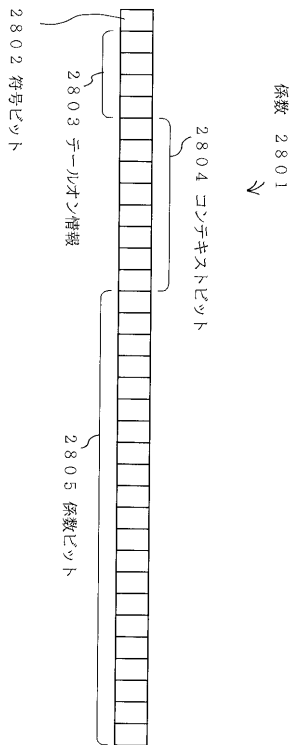
【 4 4 】



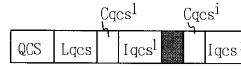
【 4 5 】



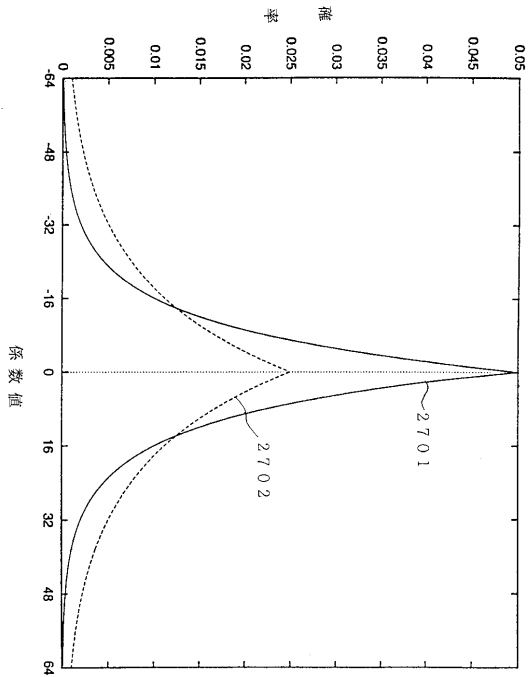
【 4 8 】



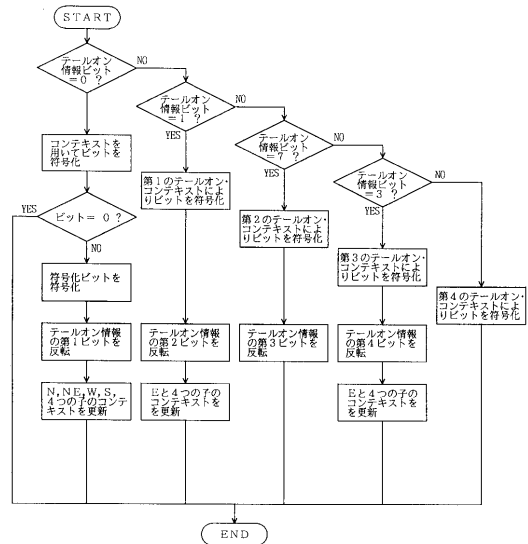
【 4 6 】



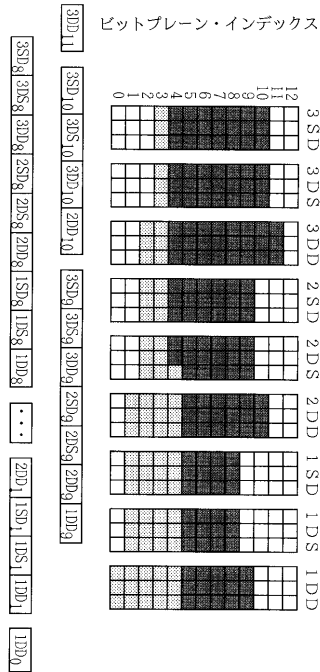
【 4 7 】



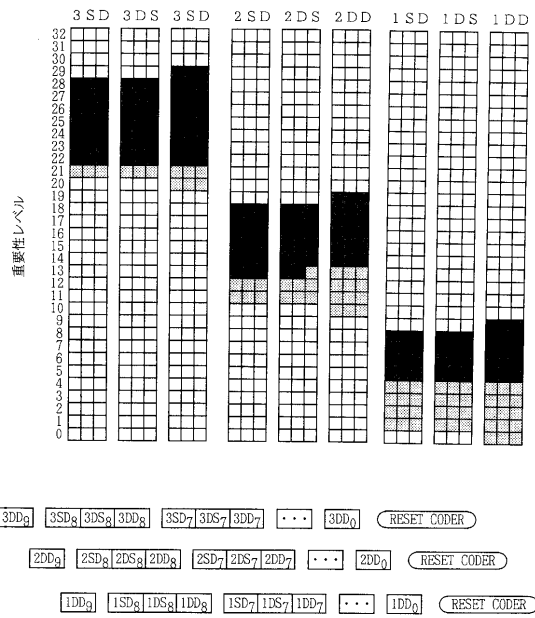
【 4 9 】



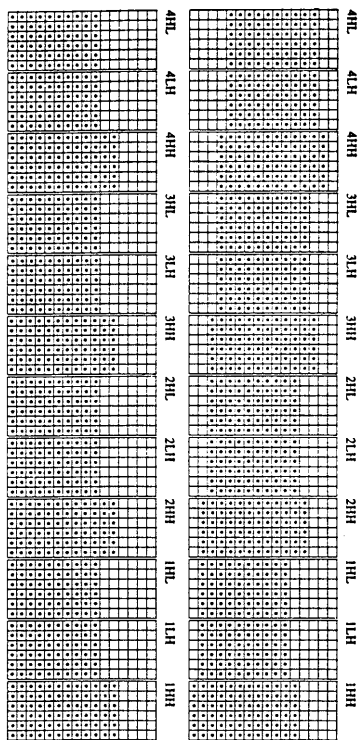
【 5 0 】



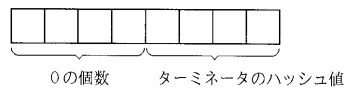
【 5 1 】



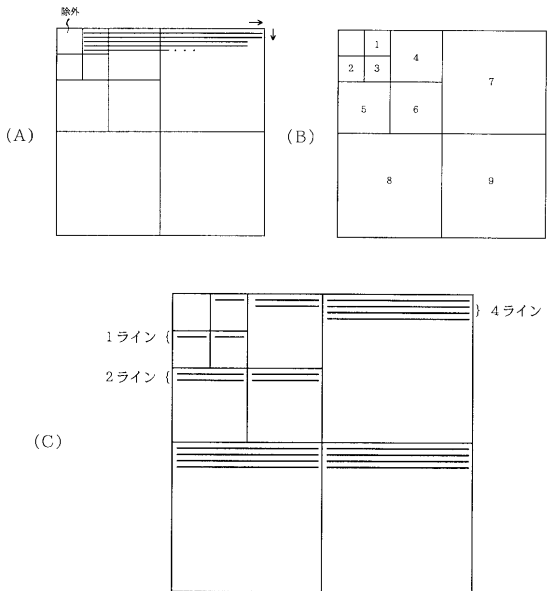
【 5 2 】



【 5 3 】



【 5 4 】







## フロントページの続き

- (72)発明者 アーマド ザンディ  
アメリカ合衆国 カリフォルニア州 94025 メンローパーク サンド ヒル ロード 28  
82 リコーコーポレーション内
- (72)発明者 マーティン ボーリック  
アメリカ合衆国 カリフォルニア州 94025 メンローパーク サンド ヒル ロード 28  
82 リコーコーポレーション内
- (72)発明者 マイケル ジェー ゴーミッシュ  
アメリカ合衆国 カリフォルニア州 94025 メンローパーク サンド ヒル ロード 28  
82 リコーコーポレーション内

## 合議体

審判長 原 光明

審判官 加藤 恵一

審判官 井上 健一

- (56)参考文献 特開平7-74966(JP,A)  
特開平6-217234(JP,A)  
特開昭60-192482(JP,A)  
Jerome M. SHAPIRO, Embedded Image Coding Using Zerotrees of Wavelet Coefficients, IEEE Transactions on Signal Processing, 米国, IEEE, 1993年12月, Vol. 41, No. 12, pp. 3445-3462  
気賀沢、他2名、“算術符号を用いたWavelet変換画像符号化”、テレビジョン学会技術報告、Vol. 18、No. 57、VIR94-52、1994年10月24日発行、37~42頁

- (58)調査した分野(Int.Cl., DB名)

H04N1/41-1/419