

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4532653号
(P4532653)

(45) 発行日 平成22年8月25日(2010.8.25)

(24) 登録日 平成22年6月18日(2010.6.18)

| | |
|---------------------|--------------|
| (51) Int.Cl. | F 1 |
| HO4N 1/41 (2006.01) | HO4N 1/41 B |
| HO4N 7/30 (2006.01) | HO4N 7/133 Z |
| HO3M 7/30 (2006.01) | HO3M 7/30 A |

請求項の数 8 (全 38 頁)

| | | | |
|--------------|-------------------------------|-----------|--|
| (21) 出願番号 | 特願2000-69353 (P2000-69353) | (73) 特許権者 | 000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号 |
| (22) 出願日 | 平成12年3月13日 (2000.3.13) | (74) 代理人 | 100076428 弁理士 大塚 康徳 |
| (65) 公開番号 | 特開2000-308060 (P2000-308060A) | (74) 代理人 | 100112508 弁理士 高柳 司郎 |
| (43) 公開日 | 平成12年11月2日 (2000.11.2) | (74) 代理人 | 100116894 弁理士 木村 秀二 |
| 審査請求日 | 平成19年2月28日 (2007.2.28) | (74) 代理人 | 100115071 弁理士 大塚 康弘 |
| (31) 優先権主張番号 | PP9186 | | |
| (32) 優先日 | 平成11年3月12日 (1999.3.12) | | |
| (33) 優先権主張国 | オーストラリア (AU) | | |

最終頁に続く

(54) 【発明の名称】 符号化方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する方法であって、

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する工程と、

c) DWTを使用して前記処理対象タイルをDWT分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と、

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、各ブロックをビットストリームに符号化する工程と、

e) 前記LLサブバンドの係数を蓄積するとともに、当該LLサブバンドの所定数の係数が蓄積されるまで、前記工程b)からe)を繰り返す工程と、

10

20

f) 前記処理対象タイルとして、蓄積された L L サブバンドの係数の前記所定数を割り当てる工程と、

g) 前記 D W T 分解の前記所定レベルに達するまで、前記工程 c) から g) を繰り返す工程と、

h) 前記 L L サブバンドをビットストリームに符号化する工程と、
を含むことを特徴とするデジタル画像を符号化する符号化方法。

【請求項 2】

前記 D W T 分解は、Daubechies 9 / 7 フィルタを使用することを特徴とする請求項 1 に記載の符号化方法。

【請求項 3】

前記 D W T 分解は、Haar フィルタを使用することを特徴とする請求項 1 に記載の符号化方法。

【請求項 4】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化方法であって、

a) 各タイルが D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また、D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回の D W T 分解で得られる L H , H L , H H から前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、

b) 前記複数のタイルの中から、D W T 分解の対象となる処理対象タイルを選択する工程と、

c) D W T を使用して前記処理対象タイルを L L 、 L H 、 H L および H H サブバンドの複数の係数を提供するように分解する工程と、

d) 工程 (c) で得られた現在レベルの L H 、 H L 、 H H サブバンドの係数をビットストリームに符号化する工程と、

e) 前記現在レベルが前記所定レベルであるかどうかを判定し、

e a) 前記現在レベルが前記所定レベルであれば、前記 L L サブバンドの係数をビットストリームに符号化し、前記工程 b) から e) を繰り返し、

e b) 前記現在レベルが前記所定レベルでなければ、前記 L L サブバンドの前回符号化されていない係数を記憶する工程と、

f) 前記現在レベルにおける前記 L L サブバンドの前記記憶された係数の数が少なくとも所定数であるかどうかを判定する工程と、

f a) 前記記憶された係数の数が少なくとも前記所定数であれば、処理対象タイルとして前記係数の前記所定数を処理対象タイルに割り当て、前記工程 c) から f) を繰り返す工程と、

f b) 記憶された前記係数の数が前記所定数より少なければ、前記工程 b) から f) を繰り返す工程と、

を含むことを特徴とするデジタル画像を符号化する符号化方法。

【請求項 5】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

a) 各タイルが、 D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また D W T 分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回の D W T 分解で得られる L H , H L , H H から前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する分割手段と、

b) 前記複数のタイルの中から、D W T 分解の対象となる処理対象タイルを選択する選

10

20

30

40

50

択手段と、

c) D W T を使用して前記処理対象タイルを D W T 分解の少なくとも一つのレベルに分解して L L 、 L H 、 H L および H H サブバンドを含む複数のサブバンドを形成する分解手段と、

d) 前記 L L サブバンドの係数の所定数を蓄積する手段と、

e) 蓄積された前記係数の前記所定数を処理対象タイルとして割り当てる手段と、

f) 前記処理対象タイルをさらなるレベルの分解度に分解するために処理対象タイルを前記分解手段に帰還させる手段と、

g) L H 、 H L および H H サブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、前記所定レベルで L L サブバンド内の前記特定のブロックサイズの少なくとも一つのブロックを蓄積する手段と、

h) 各前記ブロックをビットストリームに符号化する符号化手段と、

を有することを特徴とする符号化装置。

【請求項 6】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

各画像の部分が D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また、 D W T 分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回の D W T 分解で得られる L H , H L , H H から前記特定のブロックサイズのブロックを形成できる画素数を有するように、前記画像の少なくとも一部を記憶する手段と、

リニア変換を前記画像の前記一部の一次元および二次元それぞれに適用して、各サブバスバンドが少なくとも一つの係数を含む L L 、 L H 、 H L および H H サブバンドを提供する第 1 および第 2 フィルタ手段と、

前記 L L サブバンドの係数を所定数蓄積し、前記第 1 および第 2 フィルタ手段によってリフィルタして次のレベル分解を達成するための画像部分として、蓄積された前記係数を使用する部分バンド記憶手段と、

前記 L H , H L および H H サブバンド内の各レベルに対する前記特定のブロックサイズの前記ブロックを蓄積し、また前記所定レベルの分解レベルにおける L L サブバンド内の前記特定のブロックサイズの前記ブロックを蓄積するためのサブバンド記憶手段と、

各蓄積したブロックをビットストリームに符号化する符号化手段と、
を具備することを特徴とする符号化装置。

【請求項 7】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読み取可能な記憶媒体において、前記プログラムは前記コンピュータに、

a) 各タイルが、 D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また D W T 分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回の D W T 分解で得られる L H , H L , H H から前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、

b) 前記複数のタイルの中から、 D W T 分解の対象となる処理対象タイルを選択する工程と、

c) D W T を使用して前記処理対象タイルを D W T 分解の少なくとも一つのレベルに分解して L L 、 L H 、 H L および H H サブバンドを含む複数のサブバンドを形成する分解工程と、

10

20

30

40

50

d) L H 、 H L および H H サブバンドの各サブバンドの係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、各ブロックをビットストリームに符号化する工程と、

e) 前記 L L サブバンドの係数を蓄積するとともに、当該 L L サブバンドの係数が所定数が蓄積されるまで、工程 b) から e) を繰り返す工程と、

f) 前記 処理対象 タイルに、蓄積された L L サブバンドの前記係数の前記所定数を割り当てる割り当て工程と、

g) 前記 D W T 分解の前記所定レベルに達するまで、工程 c) から g) を繰り返す繰り返し工程と、

h) 前記 L L サブバンドをビットストリームに符号化する符号化工程と 実行させるプログラム であることを特徴とする記憶媒体。 10

【請求項 8】

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読み取可能な記憶媒体において、前記プログラムは前記コンピュータに、

a) 各タイルが、D W T 分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またD W T 分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のD W T 分解で得られる L H 、 H L 、 H H から前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、 20

b) 前記複数のタイルの中から、D W T 分解の対象となる処理対象タイルを選択する選択工程と、

c) D W T を使用して前記 処理対象 タイルを L L 、 L H 、 H L および H H サブバンドの複数の係数を提供するように分解する工程と、

d) 工程 (c) で得られた現在レベルの L H 、 H L 、 H H サブバンドの係数をビットストリームに符号化する符号化工程と、

e) 前記現在レベルが前記所定レベルであるかどうかを判定する判定工程と、

e a) 前記現在レベルが前記所定レベルであれば、前記 L L サブバンドの係数をビットストリームに符号化する工程と；工程 b) から e) を繰り返す工程と； 30

e b) 前記現在レベルが前記所定レベルでなければ、前記 L L サブバンドの前回符号化されていない係数を記憶する記憶工程と；

f) 前記現在レベルにおける L L サブバンドの前記記憶された係数が、少なくとも所定数であるかどうかを判定する工程と；

f a) 記憶された前記係数の数が少なくとも前記所定数であれば、処理対象 タイルとして前記係数の前記所定数を割り当てる割当工程と；工程 c) から f) を繰り返す工程と；

f b) 記憶された前記係数の数が前記所定数より小さければ、工程 b) から f) を繰り返す繰返工程と； 40

を実行させるプログラムであることを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、デジタル画像圧縮分野、特に画像のウエーブレット分解を実行する符号化方法および装置に関する。

【0002】

【従来の技術】

デジタル・データ圧縮、および特にデジタル画像圧縮の分野は、一時非常に注目されていたが、特に最近、離散ウエーブレット変換(DWT)に基づく圧縮技術がますます一般的にな 50

つてきている。これはDWTが画像の非冗長階層的分解を提供し、また、画像の圧縮結果が良好な定歪み統計値を提供するからである。

【0003】

一般的に、画像の離散ウエーブレット変換(DWT)は、一連の一次元DWTを用いて実行される。信号(すなわち、画像の行)の一次元DWTが、ロウパスおよびハイパスフィルタリングされ、各フィルタされた信号を2つおきに選んで実行される。この2つおきに取り出すのは、フィルタリング処理されたサンプルを2つおきに計算して保持することを意味する。たたみこみ(フィルタリング)が実行されると、フィルタは、普通は一つのサンプルであるのとは代わって、一度に二つのサンプル毎に進行する。この方法において、N個のサンプルの信号に対して、N個のDWTのサンプル、すなわち、実質的にN/2のロウパス・サンプルと実質的にN/2のハイパス・サンプルがある。

10

【0004】

【発明が解決しようとする課題】

これはランダム・アクセス・メモリ(RAM)のようなコンピュータ・メモリ内で全画像をバッファリングすることによって実行されるのが一般的で、バッファされた画像にDWTが実行され、その変換された画像が符号化される。あいにく、このアプローチには大量のメモリ、特に2000画素×2000画素或いはそれ以上を必要とし、十分な処理速度を達成しなければならない場合には、相当なバンド幅を必要とする。

【0005】

本発明は上記従来例に鑑みてなされたもので、本発明の目的は、既存の構成の一つのまたはそれ以上の欠点を実質上克服するか、ないしは少なくともこれを改善することにある。例えば、少ないバッファの容量で、複数レベルのDWT実行できる構成を提供する。

20

【0006】

【課題を解決するための手段】

本発明の第1の態様によれば、離散ウエーブレット変換(DWT)により変換して所定レベルに分解され、各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化できる複数の画素を含むデジタル画像を符号化する方法を提供することにあり、この方法は、

a) 画像を複数のタイルに分割する工程であって、各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するよう、画像を複数のタイルに分割する工程と；

30

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する工程と；

c) DWTを使用して前記処理対象タイルをDWT分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と；

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、各ブロックをビットストリームに符号化する工程と；

40

e) 前記LLサブバンドの係数を蓄積するとともに、当該LLサブバンドの所定数の係数が蓄積されるまで、前記工程b)からe)を繰り返す工程と；

f) 前記処理対象タイルとして、蓄積されたLLサブバンドの前記所定数の係数を割り当てる工程と；

g) 前記DWT分解の前記所定レベルに達するまで、前記工程c)からg)を繰り返す工程と；

h) 前記LLサブバンドをビットストリームに符号化する工程と；
を含んでいる。

50

【0007】

本発明の第2の態様によれば、

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化方法であって、

a) 各タイルがDWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、；

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する工程と；

c) DWTを使用して前記処理対象タイルをLL、LH、HLおよびHHサブバンドの複数の係数を提供するように分解する工程と；

d) 前記工程(c)で得られた現在レベルのLH、HL、HHサブバンドの係数をビットストリームに符号化する工程と；

e) 前記現在レベルが前記所定レベルであるかどうかを判定し：

e a) 前記現在レベルが前記所定レベルであれば、前記LLサブバンドの係数をビットストリームに符号化し、前記工程b)からe)を繰り返し；

e b) 前記現在レベルが前記所定レベルでなければ、前記LLサブバンドの前回符号化されていない係数を記憶する工程と；

f) 前記現在レベルにおける前記LLサブバンドの前記記憶された係数が少なくとも所定数であるかどうかを判定する工程と；

f a) 前記記憶されたLL係数が少なくとも前記所定数であれば、処理対象タイルとして前記係数の前記所定数を処理対象タイルに割り当て、前記工程c)からf)を繰り返す工程と；

f b) 記憶された前記係数の数が前記所定数より少なければ、前記工程b)からf)を繰り返す工程と；

を含む符号化方法が提供される。

【0008】

本発明の第3の態様によれば、

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する手段；

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する選択手段と；

c) DWTを使用して、前記処理対象タイルをDWT分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解手段と；

d) 前記LLサブバンド係数の所定数を蓄積する手段と；

e) 蓄積された前記係数の前記所定数を処理対象タイルとして割り当てる手段と；

f) 前記処理対象タイルをさらなるレベルの分解度に分解するために処理対象タイルを前記分解手段に帰還させる手段と；

g) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、前記所定レベルでLLサブバンド内の前記

10

20

30

40

50

特定のブロックサイズの少なくとも一つのブロックを蓄積する手段と；
 h) 各前記ブロックをビットストリームに符号化する符号化手段と；
 を有することを特徴とする符号化装置が提供される。

【0009】

本発明の第4の態様によれば、

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

各画像の部分がDWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、また、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、前記画像の少なくとも一部を記憶する手段と；

リニア変換を前記画像の前記一部の一次元および二次元それぞれに適用して、各サブバンドが少なくとも一つの係数を含むLL、LH、HLおよびHHサブバンドを提供する第1および第2フィルタ手段と；

LHサブバンドの係数を所定数蓄積し、前記第1および第2フィルタ手段によってリフィルタして次のレベル分解を達成するための画像部分として、蓄積された前記係数を使用する部分バンド記憶手段と；

LH, HLおよびHHサブバンド内の各レベルに対する前記特定のブロックサイズの前記ブロックを蓄積し、また前記所定レベルの分解レベルにおけるLLサブバンド内の前記特定のブロックサイズの前記ブロックを蓄積するためのサブバンド記憶手段と；

各蓄積ブロックをビットストリームに符号化する符号化手段と；
 を具備する符号化装置が提供される。

【0010】

本発明の第5の態様によれば、

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読み取可能な記憶媒体において、前記プログラムは前記コンピュータに、

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する工程と、

c) DWTを使用して前記処理対象タイルをDWT分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解工程と、

d) LH、HLおよびHHサブバンドの各サブバンドの係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、各ブロックをビットストリームに符号化する工程と、

e) 前記LLサブバンドの係数を蓄積するとともに、当該LLサブバンドの係数が所定数が蓄積されるまで、工程b)からe)を繰り返す工程と、

f) 前記処理対象タイルに、蓄積されたLLサブバンドの前記係数の前記所定数を割り当てる割り当て工程と、

g) 前記DWT分解の前記所定レベルに達するまで、工程c)からg)を繰り返す繰り

10

20

30

40

50

返し工程と、

h) 前記 L L サブバンドをビットストリームに符号化する符号化工程と実行させるプログラムであることを特徴とする記憶媒体が提供される。

【0011】

本発明の第6の態様によれば、

離散ウエーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読み取可能な記憶媒体において、前記プログラムは前記コンピュータに、

10

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の最小数より少ない数を有し、1回のDWT分解で得られるLH, HL, HHから前記特定のブロックサイズのブロックを形成できる画素数を有するように、画像を複数のタイルに分割する工程と、

b) 前記複数のタイルの中から、DWT分解の対象となる処理対象タイルを選択する選択工程と、

c) DWTを使用して前記処理対象タイルをL L、L H、H LおよびH Hサブバンドの複数の係数を提供するように分解する工程と、

d) 工程(c)で得られた現在レベルのL H、H L、H Hサブバンドの係数をビットストリームに符号化する符号化工程と、

20

e) 前記現在レベルが前記所定レベルであるかどうかを判定する判定工程と、

e a) 前記現在レベルが前記所定レベルであれば、前記L Lサブバンドの係数をビットストリームに符号化する工程と；工程b)からe)を繰り返す工程と；

e b) 前記現在レベルが前記所定レベルでなければ、前記L Lサブバンドの前回符号化されていない係数を記憶する記憶工程と；

f) 前記現在レベルにおけるL Lサブバンドの前記記憶された係数が、少なくとも所定数であるかどうかを判定する工程と；

f a) 記憶された前記係数の数が少なくとも前記所定数であれば、処理対象タイルとして前記係数の前記所定数を割り当てる割当工程と；工程c)からf)を繰り返す工程と；

30

f b) 記憶された前記係数の数が前記所定数より小さければ、工程b)からf)を繰り返す繰返工程と；

を実行させるプログラムであることを特徴とする記憶媒体が提供される。

【0013】

【発明の実施の形態】

以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。なお、添付図面のいづれか一つまたはそれ以上において、同じ工程および(または)特徴に同じ参照符号を付して参照するものとし、強調しない限り、これらの工程および(または)特徴は、この説明の目的としては、同じ機能または動作を有するものである。

40

【0014】

図1を参照して、4レベルまでの画像の離散ウエーブレット分散を示す。最高レベルにおいて、すなわち、レベル4において、ロウ - ロウ(L L)周波数サブバンド(またはDCサブバンド)およびハイ - ロウ(H L 4)周波数サブバンド、ロウ - ハイ(L H 4)周波数サブバンドおよびハイ - ハイ(H H 4)周波数サブバンドがある。次のレベルに下がると、そのレベルに対してハイ - ロウ(H L 3)周波数サブバンド、ロウ - ハイ(L H 3)周波数サブバンド、ロウ - ハイ(L H 3)周波数サブバンドおよびハイ - ハイ(H H 3)周波数サブバンドがある。また最低レベルにおいて、ハイ - ロウ(H L 1)周波数サブバンド、ロウ - ハイ(L H 1)周波数サブバンドおよびハイ - ハイ(H H 1)周波数サブバンドがある。

50

【0015】

各レベルは、（単一レベル）離散ウェーブレット変換(DWT) の適用によって発生される。従って、DWTを画像へ適用することは、第1レベルのDCサブバンドと、それよりも3つ高い周波数サブバンドHL1、LH1およびHH1を発生することになる。こうして、DWTが第1レベルのDC（またはLL）サブバンドに適用されて、第2レベルのDCサブバンドと、それよりも3つ高い周波数サブバンドHL2、LH2およびHH2が発生される。次に、第2レベルのDCサブバンドに更にDWTを適用することにより、第三レベルのDCサブバンドと、それよりも3つ高いサブバンドHL3、LH3およびHH3などが発生され、所望のレベルに分解される。その結果、上述の四つのレベルの分解が、（単一レベル）DWTの4つの適用、すなわち、画像への第1の適用と、上述した再帰（回帰）方法により得られたロウ - ロウ（LL）周波数サブバンドへの後続の適用とを必要とする。

【0016】

本発明の好適な実施の形態は、メモリ効率（エフィシェント）と低メモリ・バンド幅（画素当りのバイト）を実現といった点で、所定のレベルの分解を達成する。好ましい実施の形態の更なる実施は、分解されるべき画像のサイズ（1K画素×1K画素より一般的に大きい画像に対して）とは実質上無関係である。すなわち、所定レベルにおける画像の離散ウェーブレット分解は、全画像を記憶するのに必要とする量よりも小さいメモリ容量で、また、画像の幅または高さとはほぼ無関係なメモリ・バッファによって提供される。

【0017】

本発明の好適な実施の形態は、DWT分解の最大レベル数を固定にし、分解されるべき画像を複数のタイルに分割し、また、各タイルの一次元が、DWT分解の最大レベルにおけるDCブロックの対応する（第1）次元の所定サイズDCブロックを生成するのに必要な略最小画素数となるように分割することによって、十分なメモリ・バッファと低メモリ・バンド幅を達成する。さらに、各タイルの二次元は、所定サイズのDCブロックの対応する（第2）次元を生成するのに必要とする略最小画素数より少ない。好ましくは、各タイルの二次元は、画素で測定した最小長さを有していることであり、これは最大レベルにおける所定サイズDCブロックの対応する次元の（フラクション）サブ - マルチプルを生成することを必要とされるからである。

【0018】

例えば、好ましい実施の形態によれば、Daubechies 9 / 7 DWTフィルタ（すなわち、9タップ・ロウパスと7タップ・ハイパス）、および各レベルにおいて32係数×32係数（高さ×幅）のエントロピー符号化DWTブロック（アレイ）を使用して、各レベルにおいて、DWT分解の4レベルの最大まで、画像を複数のオーバラップするタイルにサブ分割する必要があり、各タイルは一次元（例えば、タイルの高さ）内で617画素となり、二次元（例えば、タイルの幅）内で71画素となる。各タイルは、分解の各反復において、ロウ - ロウ周波数部分の中間バッファリングにより4レベルにDWT分解され、所定数のDC係数が生成される。この種のタイルで約9回処理すると、DC係数の32×32ブロックがリカバーされ、エントロピー符号化される。各反復を通して、より高い周波数係数（例えば、各レベルにおけるHL、LHおよびHH）の32×32ブロックもエントロピー符号化される。

【0019】

次に、好ましい実施の形態につき、まず最初に、Daubechies 9 / 7 DWTフィルタを使用する画像の4レベルDWT分解を参照し、続いてDWT係数の32×32ブロックのエントロピー符号化を説明するが、本発明はこれに限定されるものではない。ブロックのエントロピー符号化の前に、分解レベルの異なる所定最大数、異なるフィルタおよび/または蓄積された係数の異なるブロックサイズが、本発明の範囲と概念から逸脱することなしに使用することができる。例えば、5レベルDWT分解が実行でき、および/またはDWT係数の64×64ブロックを、1つのブロックのエントロピー符号化の前に蓄積することができる。さらに、多数のウェーブレット・フィルタが「H A A R」フィルタ、「LeGa

10

20

30

40

50

「11 5/3」、「LeGall 10/18」を含めて使用することができる。この種の変形例により、次に説明するハードウェア・アーキテクチャを対応して変更してもよい。

【0020】

ここで、図2および図3を参照して、本発明の好適な実施の形態による画像のタイリングを示す。すなわち、好ましくは画像を部分的にオーバラップさせて、複数のタイルにサブ分割する。各タイルの大きさは、分解のレベル、エントロピー符号化の前に蓄積されたブロック・サイズ、および使用されるDWTフィルタの種類等の所定パラメータを固定することによって決定される。各タイルは、タイルの一つの次元の71画素に、他の次元の617画素を掛けたものからなる。図2において、タイルは水平方向に配置され、すなわち、タイルの長手側が画像の走査線と略平行にレイアウトされている。図3は、タイルが垂直方向に配置された別の構成を示しており、短手側が画像の走査線と略平行である。図2および図3の各タイルは、「タイルn」の符号が付けるられるが、ここでnは正の整数で、符号付け(ラベリング)は複数のタイルのうちの好ましい処理順序を示す。例えば、「タイル1」は「タイル2」の前に処理され、「タイル2」は「タイル3」の前に処理されるという具合である。さらに、各タイルは次に説明するように決定された近接のタイルと所定のオーバラップ域200と201を有している。

【0021】

一般的に、DWTの第2レベルにおいてサブバンドのHラインを生成するために、Daubechies 9/7 DWTフィルタを使用して、LL1サブバンドの($2H + 7$)ラインが必要で、次に、これには画像の $\{2(2H + 7) + 7 = 4H + 14 + 7\}$ ラインが必要である。2Hラインに7ラインを付加することは、Daubechies 9/7 DWTフィルタによって必要とされ、このことが「オーバラップ」200として表されている。好適には、このオーバラップは、処理されるべき2Hラインの前3ラインと、処理されるべき2Hラインの後4ラインとして分布されることである。

【0022】

数学的帰納法によれば、一つの次元において、いずれかのサブバンドのHラインを形成してレベルJにするには、

$$2^J H + (2^{J-1} \cdot \cdot \cdot + 2^1 + 2^0) \cdot 7 = 2^J H + 7(2^J - 1) \quad \text{式1}$$

の画像ラインが必要である。上式の第2項が必要とされるオーバラップである。もちろん、これは他の次元に対しても同様に適用される。例えば、Daubechies 9/7 フィルタを使用する4レベル分解($J = 4$)と、32画素幅(水平方向)W = 32のブロックを蓄積するためには、式1によれば($16 \times 32 + 7 \times 15 = 512 + 105$)幅のタイルを必要とし、ここでオーバラップ200は、105画素である。すなわち、図2中の「タイル1」と「タイル11」間のオーバラップ200は、水平方向の105画素に垂直方向の71画素($2H + 7$)を掛けたものである。

【0023】

本実施の形態では、一度に $2H + 7$ ラインを変換し、大きくなるラベル番号順に次の(オーバラップした)タイルを処理して、HL1、LH1およびHH1サブバンドの各々からHラインを生成する。これらのサブバンド部分は、このような画素(Hライン倍した)W(この例ではW = 32画素)、すなわち、2H画素列が入力されると、上述した場合のように、直ちにコード化できる。タイルの第1パス(すなわち、一つの分解レベルにおける一つのタイル)の処理の終了時点において、LL1サブバンドの1ライン当たり $\{2^{J-1}W + 7(2^{J-1} - 1)\}$ 倍したHラインをバッファする必要がある。LL1サブバンドの1ライン当たりのこの係数値は、トップ・レベル(レベルJ)におけるW幅ブロックを形成するために必要である。図2において、タイル「タイル1」が処理された後、「タイル2」がほぼ同様にして処理される。すなわち、画像の他の $\{(2H + 7) \times (2^J W + 7(2^J - 1))\}$ 画素のタイルが処理される。ここで、「タイル2」が処理されると、LL1サブバンドの他のHラインが発生され、二つのタイルが処理された後、合計2Hラインとなる。LL1サブバンドをさらに分解してLL2サブバンドのHラインを生成するためには、LL1サブバンドの($2H + 7$)ラインが必要である。「タイル2」の処理後に

は、 $2H$ ラインが生成され、「タイル3」の処理後は、 $(2H + 7)$ ラインを提供するための合計 $3H$ を提供できる。これにより次のレベルの分解を実行して $LL2$ サブバンドの H ラインを生成する。即ち、各タイル毎にDWTを行い、 $LL1$ を生成するが、次の $LL2$ を生成するために3つのタイルの $LL1$ を用いて更にDWTを実行する。この方法によれば、小さいタイル単位（小さいバッファ容量）で2レベルのDWTが実行できる。同様にして、この方法を繰り返して少なくとも $\{2^J H + 7 (2^J - 1)\}$ ラインを処理すると、分解の J 番目のレベルにおける係数の H ラインを生成することになる。即ち、レベル3（ $LL3$ ）以降の LLn の作成のためにも、その LLn の作成に必要な複数タイル分の $LL(n-1)$ を利用することにより、 H ラインの LLj を最終的に生成する。ラインが $\{2^{J-1} W + 7 (2^{J-1} - 1)\}$ 画素（またはDWT係数の長さ）であり、画像の幅または長さを必要としないことに注意されたい。次のレベルの分解を得るために各々の繰り返しにおいて、 LL 周波数のサブバンドの合成係数がバッファされ、他方で現在レベルにおいて発生されたより高い周波数（すなわち、 LH 、 HL 、 HH ）の各 $H \times W$ ブロックが符号化のためにエントロピー符号化器に送られる。10

【0024】

普通、出力デバイスおよび特に表示デバイスは、ラスタ走査順に、左から右へ、また、上から下へ画像を表示する。従って、本実施の形態の好ましい実施においては、図3に示したタイル配列となる。オーバラップするタイルは、タイルの最短次元が画像の走査ラインと平行になるように配列されるのが好ましい。好ましくは、タイルはタイルの次の列が処理される前に、画像の幅を横切って処理される。この配列は、タイルが左から右へ、また上から下へ（すなわち、ラスタ走査タイプ順）に処理できるので好ましいが、他の順序も本発明の範囲と概念から外れることなしに使用することができる。例えば、図2のタイル配列は図3のものとは別の配列である。20

【0025】

図3の配列において、各タイルはサイズ $(2^J H + 7 (2^J - 1)) \times (2W + 7)$ 画素であり、ラスタ走査順で処理される。 $J = 4$ レベルDWT分解に対し、またエントロピー符号化のために $H = 32 \times W = 32$ 係数ブロックサイズを得るために、画像の各タイルは、1タイルのライン当たり次元 $(512 + 105 = 617$ ライン $) \times (2 \times 32 + 7 = 71$ 係数（または画素））を有している。30

【0026】

図4を参照すると、図4はエンコーダ（符号化器）400のブロック図を示し、非圧縮画像401を取り入れて、画像を表す圧縮ビットストリーム402を生成する。このエンコーダ400は、所定量の外部メモリ403、好ましくはダイナミック・ランダム・アクセス・メモリDRAMと、それ自体所定量の内部メモリ405、一般的にはスタティック・アクセス・メモリSRAMを含む圧縮エンジン404とを備えている。このエンコーダ400と、特に圧縮エンジン404は必要とされる工程を実行し、非圧縮画像401から圧縮画像ビットストリーム402を生成し、リニア変換（DWT）の実行と、その変換係数をエントロピー符号化するエントロピー符号化器を含んでいる。さらに、圧縮エンジン404は単一チップとして設けられ、そのチップを通してマスター画素クロック信号clkを使用するのが好ましい。40

【0027】

非圧縮画像401は、エンコーダ400の外部DRAMによりバッファリングされ、かつ、圧縮エンジン404によって制御される。外部DRAMは、これまでに説明したタイル（ 617×71 画素又は係数）に略対応する非圧縮画像の一部をバッファするのに使用されるのが好ましい。圧縮エンジン404は、各ラインに沿った71画素または係数に対応するデータの短いバーストを読み取り、タイルをジグザグ順に上から下に走査する。

【0028】

図5はジグザグ順500を示し、この順に各タイルが走査され、図3の配列よりも詳しいイラストを画像502の部分で示す。これまでに説明したように、タイルの幅は $(2W + 7)$ であり、 $W = 32$ を所望のブロック幅とすると、タイル幅（またはタイル内のライン

10

20

30

40

50

の長さ)は71画素(すなわち、64+7、7は水平方向のオーバラップ201)となる。従って、圧縮エンジン404は、617ライン、すなわち、H=32としたとき、(2^JH+7(2^J-1))ラインが読み取られるまで、同時にジグザグ順で走査されてタイルの1本のラインを一つ読み取るのが好ましい。すなわち、前回の説明部分は、近接するタイル間のオーバラップ域200の垂直方向の寸法を構成している。例えば、四つのレベルDWTに対し、オーバラップ域200は、タイルの長さ方向における105画素に幅(71画素)を掛けたものである。より高い周波数要素(LH、HL、HH)の各32×32(H×W)変換ブロックが発生されると、このブロックは中間バッファリングなしにエントロピー符号化される。したがって、好ましい実施の形態において、水平方向における二つの近接するタイル(例えば、「タイル1」と「タイル2」)間のオーバラップ域201は、略7画素幅にタイルの長さを掛けたもので、垂直方向の二つの近接するタイル(例えば、「タイル1」と「タイルi」)間のオーバラップ域200は、略105画素長さにタイルの幅を掛けたものである。

次に図6は、図4の圧縮エンジン404をより詳細に示す。圧縮エンジン404は:アドレス・バス603を介してDRAMに伝送されるメモリ・アドレスに応答して、外部DRAM403からの602データを受取るリオーダ(再配列)・バッファ・モジュール601と;制御状態マシン・モジュール604と;垂直フィルタ605と;水平フィルタ・モジュール606と;サブバンド・バッファ607と;部分バンド・バッファ608と;エントロピー符号化ユニット(符号化器)609を含んでいる。

【0029】

リオーダ・バッファ・モジュール601は、外部DRAM403に対する読み取りと書き込みを制御するとともに、DRAM403から受信したデータを組立て、所定順序で垂直フィルタ605で読み取りできるようにする。このリオーダ・バッファ・モジュール601は9バイト(72ビット)のデータを一度に、垂直フィルタ・モジュール605に送る。垂直フィルタ・モジュール605は、第1次元(図3のタイルの垂直方向)で、ロウパスおよびハイパス・フィルタリングを実行する。垂直フィルタ・モジュール605からの二つの出力611と610があり、それぞれデータのロウパスおよびハイパス・フィルタリングに対応している。各出力610および611は、水平フィルタ・モジュール606内の分離水平フィルタにバスされ、ここで同一のフィルタ操作が第2次元(図3のタイルの水平方向)において実行される。水平フィルタ・モジュール606は、LL、LH、HLおよびHHサブバンド信号に対応する四つの出力信号を提供する。高い周波数成分(すなわち、LH、HLおよびHH)を含むサブバンド信号が、エントロピー符号化器609によって符号化される前に、サブバンド・バッファ・モジュール607内で一時的にバッファされ、613に符号化ビットストリームを出力する。LLサブバンド信号が部分バンドバッファ608にバスされ、ここで、より大きいブロックに合成され、後続の変換レベルのために垂直フィルタ・モジュール605に帰還される。最終レベル(例えば、レベル4)においてのみ、LLサブバンド信号が出力ビットストリーム(出力613を介して)に符号化される。制御状態マシン・モジュール604が、圧縮エンジン404の動作を管理し、適切なデータが正しい場所に記憶されるか、また、その動作が正しい時刻に正しいデータで実行されるかどうかを確認する。

【0030】

次に図7を参照して、図6のリオーダ・バッファ・モジュール601をより詳しく説明する。このリオーダ・バッファ・モジュール601は、外部メモリ403から受信した中間生データ602を記憶するランダム・アクセス・メモリ(以下、リオーダRAMという)700と、外部DRAMサブモジュール403を制御するためのDRAMコントローラを含むリオーダ・アドレス・ジェネレータ・サブモジュール701とを含んでいる。data_val_id(データ・ヴァリッド)信号702は、(生)の画素データ401をDRAM403に供給する外部デバイスから供給される信号であって、DRAMデータ・ピン上の現在データが有効であることを示している。data_ack信号703は、データがDRAM403に読み取られたことを確認する。外部DRAM403が、その入力を外部画素供給源(図示

10

20

30

40

50

省略)から直接取り入れる。リオーダ・アドレス・ジェネレータ701は、アドレス704を決定してリオーダRAM700に供給し、そのRAM700への書き込みを制御する。

【0031】

リオーダRAM700からのデータ・アウト(Data Out)(DO)ピンが、同じリオーダRAM700のデータ・イン(Data In)(DI)ピンに、8ビットシフトして接続されている。すなわち、DO(n+7:n)ピンがDI(n+15:n+8)ピンに接続され、ここで、nは低い方のビット数を表している。例えば、72ビット・ワード・データ・アウトの少なくとも8個の重要なビットを表すDO(7:0)ピンが、72ビット・ワード・データ・インの第2の少なくとも8個の重要なビットに対応するDI(15:8)に接続されている。データが、読み取・修正・書き込(リード・モディファイ)技術を使用してリオーダRAM700に書き込まれる。72ビット・ワード・データ・インの第2の少なくとも8個の重要なビット、すなわち、DI(7:0)ピンへのバイト入力が外部DRAM403から入り(DRAM Data)、一方残りの入力データ・ワードが上述したシフトされた様でリオーダRAM700の出力から入力されている。従って、リオーダRAM700は、大きいバイト幅をアドレス可能なシフト・レジスタのように、9ワード深さと71ワード幅で作動する。データが、画像の走査線(すなわち、タイルのライン)に沿って71ワードのバースト内で外部DRAMから読み取られる。画像データの二つのライン(71バイト×2)が、垂直フィルタ・モジュール605によって使用されるサンプルの各列(ライン)に対してリオーダRAM700に書き込まれる。これはDWによって必要とされる垂直方向内の画像データのダウン・サンプリング(1/2に減らすこと)を構成ないし提供する。

【0032】

上述したように、リオーダRAM700からのリオーダ・データが、ro_dataバス705を介して、一度に9バイト(72ビット)が垂直フィルタ・モジュール605に供給される。

【0033】

図8は図7のリオーダ・バッファ601の構成を表したものである。好適な実施の形態のリオーダ・バッファ601は、71画素幅と9ラインで構成されている。このバッファ601は、図7において矢印801で示したように左から右に書かれる。各ラインが左から右に書かれると、1ライン分の画素が一度にライン分プッシュダウンされる。すなわち、バッファの9ラインが書込まれると、(9×71=)639画素がバッファを占有し、639画素の最初(左上)の画素が、バッファの9番目のライン804の最初の位置にあるバッファのセル802の位置のバッファに入り、一方639画素の最後(右下)の画素が最初のライン805上のバッファのセル803の位置のバッファに入る。メモリ700(リオーダRAM)が72ビットの71ワードとして編成される。ここで72ビットは垂直方向のディメンジョン(寸法)である9画素を表している。9ラインが同じ71アドレスに繰り返して(一度に8ビット)書き込まれると、全バッファ601が満たされる。639画素の最終71バイト(画素)がトップ・ラインに書き込まれると、読み取・修正・書き込サイクル(全72ビット)が使用されている間、データが読み取られて垂直フィルタ・モジュール605にロードされる。一度、リオーダ・バッファ601が満たされると、データがバッファ601にロードされる画像データのどの第2ラインからでも垂直フィルタ・モジュール605によって使用される。これは垂直方向におけるサブ・サンプリングとなる。

【0034】

垂直方向におけるサブ・サンプリングは次のようにして達成される。最初に、垂直フィルタ・モジュール605がリオーダ・バッファの9ラインを処理する。このとき、9タップ・フィルタの内の1番目のタップが、リオーダ・バッファの最終ライン805からのデータを使用するようにアレンジされ、一方、フィルタの最終タップがリオーダ・バッファ601の第1ライン804からのデータを使用して、例えば、部分バンド・バッファ608内に記憶されることになるロウ・パス・フィルタされたデータの1ラインを生成する。当

然、1番目のタップと最終タップの間のタップは、リオーダ・バッファ601の対応するラインからのデータを使用する。次に、リオーダ・バッファ601の各ラインのデータはプッシュダウンされ、リオーダ・バッファ601の最終ライン804は除去され（バッファから押し出され）、各先行ラインからのデータがリオーダ・バッファ601の次のラインに書き込まれる。すなわち、バッファの8番目のライン806からのデータが、9番目のライン804に、また、バッファの7番目のラインからのデータが8番目のライン806に書き込まれ、6番目のラインからのデータが7番目のラインにというようにして書き込まれる。リオーダ・バッファ601の初頭ライン805において、新しいラインデータが外部DRAM403から入力される。このプッシュダウンが、リオーダ・バッファ601に入力された全てのラインに対して繰り返される。こうして、第2ラインのデータがリオーダ・バッファ601に入力された後、垂直方向内のサブ・サンプリングが、リオーダ・バッファの9ラインの読み取りと処理によって実行される。このリオーダ・バッファは、タイルの開始時点で完全に再充填され、全画像が処理されるまで、上述の処理が繰り返される。
10

【0035】

画像の略71ラインが外部メモリ403から検索され、かつ、リオーダ・バッファ601に伝送された後、サブバンド・バッファ607がエンコーディングを開始するための十分な変換係数を含んでいる。好適な実施の形態において、サブバンド・バッファ607がいっぱいになり、リオーダ・バッファ601がさらに更新される前に、係数が符号化される必要がある。より高いレベルの係数がフィルタリングされている間、次のレベルのフィルタ操作の準備のために、新しいデータがリオーダ・バッファにロードされる。
20

【0036】

リオーダ・アドレス・ジェネレータ701は、リオーダ・バッファ601のために、単純に0から70までの周期でアドレスを発生するが、読み取り中、図5を参照してこれまでに説明したように、ジグザグに外部DRAM403をアドレスする。このリオーダ・アドレス・ジェネレータ701は、画像を外部DRAM403に書き込み中、ラスタ走査順の制御もする。外部DRAM403は、画像の少なくともフルに617幅のラインを記憶するのに十分な大きさであるのが好ましい。実際には、十分なバンド幅はデュアル・ポートDRAMで実現でき、あるいはオプションとして、ビデオ・ランダム・アクセス・メモリ(VRAM)が外部メモリ403として使用できる。
30

【0037】

図9は図6の垂直フィルタ・モジュール605をより詳細に示す図である。この垂直フィルタ・モジュール605はマルチプレクサ901を備えており、レベル1の分解を実行するときのリオーダ・バッファ601からのデータか、又は後続の分解レベル中、部分バンドバッファ608からのデータのいずれかを選択する。リオーダ・バッファ601から受信されたデータの9バイト×8ビットが16ビットの9ワードに変換される。これは未使用ビットをゼロにすることによって、部分バンド・バッファ608からのデータと一致する。マルチプレクサ901は、制御状態マシン・モジュール604によって制御される。一般的に、1画素データ当たり8ビットのカラー・チャネルが16ビットの精密DWT係数に変換され、これによって情報の最小量が係数値に対する丸め効果のために失われる。
40

【0038】

さらに、垂直フィルタ・モジュール605は垂直ロウ・バス・フィルタ902と、垂直ハイ・バス・フィルタ903を含み、これらはマルチプレクサ901の選択に基づいてデータを受信する。垂直ロウ・バス・フィルタ902は、9×16ビットのサンプルを得て、データに対してマトリックス操作（乗算・蓄積）を実行し、入力の全9ワードから16ビットの結果(vfl_data)を生成する。垂直ハイ・バス・フィルタ903は、7×16ビット・サンプルのみを必要とし、そのマトリックス操作を実行し、再度16ビットの結果(vfh_data)を生成する。これまでに説明したように、本実施の形態のフィルタはDaubechies 9/7 DWTに基づいているが、他のウェーブレット・フィルタも使用できる。

【0039】

図10は、図6の水平フィルタ・モジュール606のより詳しい説明図である。この水平フィルタ・モジュール606は、垂直フィルタ・モジュール605からのデータを処理する二つのほぼ同様のセクション1004と1005を含んでいる。各セクションは9ステージ・シフト・レジスタ1001、水平口ウ・パス・フィルタ1002および水平ハイパス・フィルタ1003を含んでいる。データはまず最初に、垂直フィルタ・モジュール605から受信された最大9個の連続するサンプルをバッファする、一対の9ステージ・シフト・レジスタを通過して同時に処理される。一方、二つのセクション1004と1005はほぼ同じ方法でデータを処理するが、各セクションのためのデータ源は異なる。一つのセクション1004が、垂直口ウ・パス・フィルタ信号vfl_dataを受取り、一方、他方のセクション1005が垂直ハイ・パス・フィルタ信号vh_dataを受取る。

10

【0040】

各セクションの水平口ウ・パス・フィルタ1002が 9×16 ビット・サンプルを取り込み、そのデータにマトリックス操作（乗算・蓄積）を実行し、入力の全9ワードの各々の結果から16ビットを生成し、セクション1004と1005それぞれからlh_data信号1006とhl_data信号1007を生成する。各セクションの水平ハイ・パス・フィルタ1003は 7×16 ビット・サンプルのみを必要とし、マトリックス操作を実行し、再度各ハイ・パス・フィルタ1003が16ビット結果を生成する。セクション1004の水平ハイ・パス・フィルタが信号lh_data11008を提供し、セクション1005の水平ハイ・パス・フィルタが信号hh_data11009を提供する。その結果得られたサブバンド・データ、すなわち、LL、LH、HLおよびHHサブバンドの係数に対応する信号lh_data、lh_data、hl_data、およびhh_data信号がサブバンド・バッファ・モジュール607に送られる。

20

【0041】

9ワード・サンプルの処理後、両9ステージ・シフト・レジスタがデータを2だけシフトし、水平方向にサブ・サンプリング（1/2に減少）のためにアカウントする。

【0042】

DWT分解の現在レベルにおけるLLサブバンドの係数に対応するlh_data信号1006も、分解の続くレベルのために部分バンド・バッファ・モジュール608に送られる。

30

【0043】

図6のサブバンド・バッファ・モジュール607を図11により詳しく示す。このサブバンド・バッファ607は、LL、LH、HLおよびHHサブバンドの各々から所定数の係数を記憶する所定数のランダム・アクセス・メモリ（RAM）1101と；サブバンド・バッファ・アドレス・ジェネレータ・サブモジュール1102とマルチプレクサ1103を含んでいる。サブバンド・バッファ・アドレス・ジェネレータ・サブモジュール1102は、サブバンドRAM1101にアドレスを供給するとともに、制御状態マシン・モジュール604の制御下でRAMへの書き込みの制御もする。

【0044】

マルチプレクサ1103は、エントロピー符号化ユニット（エントロピー符号化器）609によって符号化するために四つのサブバンドの一つから、データと係数を選択する。

40

【0045】

図12を参照して、図11のサブバンド・バッファのデータ構成ないし編成の態様を説明する。サブバンド・バッファは、 32×32 個の係数からなる四つのバンク1201を保有しており、各係数は16ビット値で表わされる。従って、サブバンドRAMは合計64キロビットの容量で、1ワード64ビットで1024ワードで構成されており、各メモリ位置で並行して、各サブバンドからの1つの係数をストアする。即ち、現レベルで、各サブバンドからの1つの係数、計四つの係数が、それぞれ64ビット・ワードの16ビットとして表わされる。バッファ1101、特に各バンク1201が、図12に示したジグザグ・パターン1202によって示された上から下へのラスター操作順に満たされる。エンコーディング中、エントロピー符号化ユニットにおいて、サブバンドRAM1101が4分木(quadtree)方式で読み取りされる。

50

【0046】

図13を参照して、図6のエントロピー符号化ユニット609をより詳しく説明する。このエントロピー符号化ユニット609は、ビットプレーン・コンバータ・モジュール1301と16個のビットプレーン・メモリを含むビットプレーン・バッファ・サブモジュール1302と；複数のビットプレーン・ツリー・ビルダー・ユニット1303と対応するビットプレーン・ツリー・バッファ・ユニット1304と；ビットプレーン・エンコーダ1305と；ビットプレーン・エンコーダ1305によって使用されるデータの中間記憶のための複数の中間バッファ1306とを備えている。サブバンドRAM1101に記憶されたデータがサブバンド・バッファ・モジュール607からクワドトリー順に読み取られる。ビットプレーン・コンバータ・モジュール1301は、各ビットプレーンのデータを分離するとともに、ビットプレーン・バッファ・サブモジュール1302内の16個のビットプレーン・メモリの各々に書き込む。データはまた、処理されたビットプレーン・ツリー構造情報を、対応するビットプレーン・ツリー・バッファ1304に書き込む複数のビットプレーン・ツリー・ビルダーによって走査される。ビットプレーン・ツリー1304とビットプレーン・バッファ1302の各々からのデータが、ビットプレーン・エンコーダ1305によって使用され、ビットストリームの発生を実行して、これが（バス613を介して）出力される。ビットプレーン・エンコーダ1305は制御状態マシン・モジュール604によって同期化（制御）される。中間バッファ1306は、重要な係数のリスト（LSC）を一時的に記憶するバッファと；重要な係数（LIC）のリストを一時的に記憶するバッファと；域バッファの第1リストと域バッファの第2リストとしての二つのバッファとして実現できる古い重要な域（OIR）リストを一時的に記憶するバッファ（図13では不図示）を含んでいる。さらに、エントロピー符号化ユニット609は、サイン・ビット・バッファ（図13では不図示）を含み、これはビットプレーン・コンバータ・モジュール1301の出力と、サイン・ビットをビットストリームに符号化するビットプレーン・エンコーダ1305とに接続されている。エントロピー符号化ユニットの一例を本明細書中の、見出し「エントロピーエンコーダ」で後ほど詳細に説明する。

【0047】

本発明の好適な実施の形態を、上述したクワドトリー・ビットプレーン・エンコーディングを実行するエントロピー符号化ユニットを参照して説明するが、他のエントロピー符号化器も本発明の範囲と概念から外れることなく使用することができる。例えば、ビットプレーン演算コーダが変換係数を符号化するのに使用でき、または隨意に、一部クワドトリー・エンコーディングとDWT係数の一部演算コーディングを実行するハイブリッド・エンコーダを使用できる。

【0048】

図14を参照して、図6の部分バンド・バッファ・モジュールをより詳しく説明する。部分バンド・バッファ・モジュール608は：分解の各レベルに対するLLサブバンド・データ（すなわち、図6の水平フィルタ・モジュール606によって発生されたll_data1006）を記憶するRAM1401と；部分バンド・バッファ・アドレス・ジェネレータ・サブモジュール1402と、バ렐・シフト・レジスタ（バ렐・シフター）1403とを含んでいる。部分バンド・バッファ・アドレス・ジェネレータ1402がアドレスを部分バンドRAM1401に供給するとともに、制御状態マシン・モジュール604の制御下で、そのRAMへの書き込みを制御する。

【0049】

部分バンド・バッファは、十分なサンプルが分解の次のレベルをフィルタするのに利用可能になるまで、圧縮の各レベルからLLサブバンド・データを保持する。一旦、十分なサンプルがいずれか一つのレベルにおいて利用可能になると、データがバ렐・シフター1403を介して垂直フィルタ・モジュール605に読み取られ、分解の次のレベルがフィルタされる。概念的に、RAM1401は三つの別々のバッファを含み、各々100個のセル幅を有し、各セルは16ビット値を記憶することができる。これら三つの別個のバッ

ファの内の第1バッファは305行(row)(ライン)深さで、第2バッファは149行の深さで、第三バッファは71行の深さであり、総計525行(ライン)になる。しかし、垂直フィルタ・モジュール605が、Daubechies 9/7フィルタによって必要とされる9個の値を取るので、行(ライン)の合計は9の倍数であるのが望ましい。従って、余分の6行のセルがバッファに付加され、結果として合計531行(すなわち、9の59倍)になる。本実施の形態において、この結果は少なくとも $59 \times 100 \times 144 = 531 \times 100 \times 16$ ビット、すなわち、約0.81メガバイトのメモリ・サイズになる。

【0050】

バレル・シフト・レジスタ1403が、データの單一クロック・アライメントを垂直フィルタ・モジュール605によって必要とされる所望の行にすることを許容する。部分バンドRAM1401から読み取って垂直フィルタ605のための單一ワードを生成するために所定のメモリ位置が必要となる。必要とされる單一ワードが9ラインの境界線に沿って包括される場合、二つの近接するメモリ位置が読み取られる必要がある。これは一般的に、9ライン・セット(各セットは9ラインからなる)の内8本に発生する。この場合、データは、両ワードが読み取りされ、全ての必要とされるビットが利用可能、かつ、適切に配置されるまで、バレル・シフター1403内にラッチされる。

10

【0051】

図15を参照して、部分バンド・バッファ608のデータ編成の態様を説明する。部分バンド・バッファ608は、100係数幅と525係数高さと、上述したように、これに100係数幅の行(図15では不図示)につき、余分の6を加えたのもである。部分バンド・バッファ608は、タイルの少なくとも一つのレベルDWT分解が実行された後、データ(係数)に書き込まれる。従って、分解の1レベル後で、画像のタイル($2^J H + 7 (2^J - 1) \times (2W + 7)$)が、係数のタイルを少なくする。すなわち、分解の第1レベルが 305×32 係数のタイルまで少なくなった後、 617×71 画素タイルになる。好適な実施の形態の部分バンド・バッファ608は、100係数幅において、少なくとも幅方向に三つの32係数幅タイル1501と4係数幅ブロック1502を収容するように設計される。各32係数幅のタイルが、図15に示すように一度に1レベルで、左から右に、また、上から下の走査ライン順に書き込みされる。すなわち、第2レベル分解(各々、長さ $256 + 49 = 305$ 係数)のための三つの32係数幅のタイル1501が、各々ジグザグ順に満たされる。4係数幅のブロック1502が、少なくとも最初のタイル処理のために、最初の32係数幅タイル1501(以後、「パッディング(padding)」ないし「パッデッド(padded)」という)の係数の最初の4カラムの鏡像で満たされる。このパッディングは普通、画素境界線外の画素データが存在しない画像の境界線のためにアカウントするように実行される。他のパッディング技術が、画像境界線のためにアカウントするのに利用できる。例えば、最も簡単な、しかしつつある必要はない、オプションが、画像の境界線をゼロ(0)でパッドするものである。

20

【0052】

第2レベルの分解のための係数が、垂直フィルタ605と水平フィルタ606モジュールを介して処理されると、第三レベルの分解のための係数が、次のレベルの32係数幅のタイル1504と長さ方向($128 + 21 = 149$)係数に蓄積される。再度、最初に4係数幅ブロック1503が対応する(近接)タイル1504の係数でパッドされる。第三レベル分解の係数が垂直フィルタ605と水平フィルタ606モジュールを介して処理されると、4レベル分解のための係数が次のレベル32係数幅のタイル1506に蓄積される。再度、最初に4係数幅のブロック1505が、対応する(近接)タイル1506の係数でパッドされる。画像の境界線において、4係数幅・ブロック1502、1503および1505がパッドされるべきとき以外の場合において、4係数幅ブロックが先行タイルの最終4カラムを含み、このレベルで処理される。

30

【0053】

各レベルにおいて4係数幅ブロックが、水平線をフィルタするためのオーバラップとして使用される。1レベルにおいて、一つの完全な32係数幅タイルを発生するために、先行

40

50

レベルにおいて 71 係数幅が必要である。第 2 レベルの分解のための 305×71 係数からなる図 15 の隠れたエリア 1507 には、第三レベルの分解のための 149×32 のタイルを発生することが要求される。この隠れたエリア 1507 内の係数は、走査ライン順に、1 ラインの左から右へ 71 カラム分、また上から下へ本読み取られる。一度、部分バンド・バッファ 608 が次の（高い）レベルで所望の量（例えば 71 カラム）まで満たされると、現在レベルにおけるカラム・アドレスが 64 カラムだけ右に増大し、アドレスが 100 番目のカラムを越えると第 1 カラムを包み込む。これは、例えば、現在レベルにおいて、係数の次の 71 カラムの処理が、バッファ 608 の 64 番目のカラム 1508 で開始され、バッファ 608 の第 1 カラムの近くまで包み込まれる。カラム 64 からカラム 68 までが、ここで 4 係数幅ブロック 1509 のオーバラップの必要性を提供し、データのもう一つの 71 カラムを処理する必要がある。10

【0054】

部分バンド・バッファ 608 の各レベルがフィルタされると、フィルタ（LL サブバンド・係数）の結果が、図 15 に示した部分バンド・バッファ 608 の次の高いレベルに書き込まれる。最も高いレベルがフィルタされてしまうと（例えば、図 15 の「レベル 4」）、得られた LL サブバンド・係数が、これらの係数を部分バンド・バッファ 608 にさらに書き込みせずに、サブバンド・バッファ 607 に書き込み、さらにフィルタせずにストレートに符号化される。

【0055】

部分バンド・バッファの読み取りの際に、 $9 \times 16 = 144$ ビットが一度に読み取られる。これらの 9 ワードはバッファの單一カラムからの 9 個の垂直係数を表している。このバッファから完全に読み取られるライン間で、ライン・アドレスが 2 だけ増大するので、144 ビットストリームがバレル・シフタを通り、このデータと垂直フィルタへのデータ・バスに整列される必要がある。さらに、続いて 2 ワードを読み取り、一つの 144 ビット・ワードを得て、レジスタ内に第 1 ワードの必要とするビットを保持することによってワードの二つの部分を構築し、他方で第 2 アクセスから必要とされるビットがフェッチされる。ライン・アドレスの一致が 9 の倍数であるときのみ、一つのフェッチだけが必要である。一度に 144 ビットをフェッチできるようにするために、メモリは物理的に 144 ビット幅にしなければならない。余分なデコーディングが、メモリ・アレイに書き込み可能な形態にする必要がある（従って、ちょうど 16 ビットが同時に書き込みできるようにする）。20

【0056】

図 16 は、図 6 の制御状態マシン・モジュール 604 のための流れ図を示す。この制御状態マシン・モジュール 604 は、各モジュール 601、604 ~ 609 の機能を制御すること、およびデータを各モジュールにモジュール機能と同期させて（すなわち、正確な時間で）正しくルーティング（経路指示）する。30

【0057】

画像圧縮の開始 1601 において、「レベル」変数が 1602 で 1 に設定される。これは一般的に実行されるべき分解の第 1 レベルを表している。決定ブロック 1603 は、DWT 分解の現在レベルが第 1 レベルにあるかどうかを判定する工程に入る。この決定ブロック 1603 が真（YES）の返答であれば、DRAM（図 4）に記憶された画像の最初の 9 ラインがフィルタされ（1604）、サブバンド・データの第 1 ラインが生成される。入力データがレベル 1 のために外部 DRAM 403 から取り出される。そうでなければ、決定ブロック 1603 が、偽（NO）の返答をし、ブロック 1605 にこれを送って、ここでデータが内部部分バンド・バッファ 608 から取り出され、フィルタ（次のレベルに DWT 分解）される。十分なデータが部分バンド・バッファ 608 から取り出され、次のレベルの DWT 係数のタイル内のラインを生成する。これらの結果はこれまでに説明したように対応するレベルのためのロケーションの部分バンド・バッファ 608 に記憶される。40

【0058】

データの 71 ラインがフィルタされた後、サブバンド・バッファ 607 の 32 ラインがいっぱいにされる。この時点において、流れ図 1600 はサブバンド・バッファ 607 がいっぱいであるかどうかをチェックする。このチェック 1606 が偽 (NO) の返答をすれば、処理流れは決定ブロック 1603 に戻り、決定ブロック 1603 を参照して上述した処理を続行する。チェック・ブロック 1606 が真 (YES) を返答すれば、さらなる決定 (ないしチェック) ブロック 1607 に進み、データがフィルタされるべきレベルのためのオーバラップ域内にあるかどうかがチェックされる。この域になければ、ブロック 1607 が偽 (NO) を返答し、サブバンド・バッファ 607 内に記憶された三つのサブバンド LH、HL および HH が符号化される (1608)。これら三つのサブバンドのエンコーディングに続いて、決定ブロック 1609 に進み、分解の現在レベルが最大レベル (すなわち、レベル 4) であるかどうかがチェックされる。現在レベルが 4 であれば、決定ブロック 1609 が真 (YES) を返答し、サブバンド・バッファ 607 内の LL サブバンドも符号化され (1610)、流れ処理が決定ブロック 1611 において続行される。さらに、決定ブロック 1609 において、偽 (NO) が返答されれば、サブバンド・バッファ 607 内の LL サブバンドが符号化されずに、流れ処理が決定ブロック 1611 で続行される。

【0059】

決定ブロック 1607 を介して、ブロック 1611 への他のエントリがあり、ここで決定ブロック 1607 が真 (YES) を返答してくる、即ち、処理されるべきデータが、近接するタイルのオーバラップ域 201 または 200 内にある場合にブロック 1611 に進む。この決定ブロック 1607 において実行されたチェックは、オーバラップ域の符号化を二度することを回避するためで、決定ブロック 1607 が真 (YES) を返答したときに、決定ブロック 1611 に進行する。

【0060】

決定ブロック 1611 において、部分バンド・バッファ 608 (図 6) のレベル 2 において、(256 + 105) 行で 71 カラムの完全な係数のセットを有しているかどうかがチェックされる。もしそうであれば、「レベル」変数が 1612 で 2 にセットされ、処理が決定ブロック 1603 において再開され、上述したように工程 1605 から 1611 までが続行される。

【0061】

レベル 2 で部分バンド・バッファがまだ一杯にされていなければ、即ち、完全なセット (すなわち、71 × 361 係数からなるレベル 2) が部分バンド・バッファ 608 になれば、偽が決定ブロック 1611 によって戻され、チェック (決定) ブロック 1613 に進む。チェック・ブロック 1613 が、部分バンド・バッファのレベル 3 において、(128 + 21) 行の 71 カラムの完全な係数セットが存在するかどうかをチェックする。もしそうであれば、「レベル」変数が 1614 で 3 にセットされ、処理が決定ブロック 1603 で再開され、部分バンド・バッファ 608 のレベル 3 から回収されたデータで、上述したように工程 1605 から 1613 が続行される。

【0062】

レベル 3 で部分バンド・バッファがまだ一杯にされていなければ、即ち、完全なセット (即ち、71 × 149 係数からなるレベル 3) が部分バンド・バッファ 608 になれば、偽が決定ブロック 1613 によって戻され、チェック (決定) ブロック 1615 に進む。チェック・ブロック 1615 が、部分バンド・バッファのレベル 4 において、(64 + 7) 行の 71 カラムの完全な係数のセットが存在するかどうかをチェックする。もしそうであれば、「レベル」変数が 1616 で 4 にセットされ、処理が決定ブロック 1603 で再開され、部分バンド・バッファ 608 のレベル 4 から回収されたデータで、上述したように工程 1605 から 1615 が続行される。

【0063】

レベル 4 で部分バンド・バッファがまだ一杯にされておらず、従って、偽 (NO) が決定ブロック 1615 によって返答されると、チェック (決定) ブロック 1617 に進む。ブ

10

20

30

40

50

ロック 1617において、全画像が完全に処理されたかどうかがチェックされる。この場合に、決定ブロック 1617が真(YES)を返答し、流れ処理が終了する(1618)。そうでなければ、決定ブロック 1617が偽(NO)を返答し、処理がブロック 1602で再開され、「レベル」変数に1を割り当てて、上述したように工程 1602から 1617が再度、外部DRAM403から取り出されたデータに対して処理が続行される。

【0064】

説明した実施の形態において、9本のソース・ラインが係数の第1ラインを発生するのに必要とし、一方、2本のソース・ラインが次のおよび係数の各続くラインを発生するのに必要となる。71画素(または係数)の最小71ラインが32係数の32ラインを形成する必要がある。部分バンド・バッファ 608が、次のレベルを処理するために、係数の三つのタイルを最初にロードする。次のレベルにおいて発生されるべき全ての連続するタイルに対して、入力の二つの付加的なタイルのみが必要である。これは部分バンド・バッファ 608が既に前回のロード・データから少なくとも一つのタイルを含んでいるからである。本実施の形態において、71カラムの係数が必要であり、部分バンド・バッファにおいて、現在レベルのために次のレベルフィルタする必要がある。すなわち、フィルタ操作は前回処理されたタイルから4カラムの係数で開始され、二つの付加的(完全な)タイル(各々32カラム)と、71カラムのタイルを形成するのに処理されるべき次のタイルからの3カラムとを使用する。随意に、部分バンド・バッファ 608は、図15に示したような単一のバッファ・メモリとしてバッファ 608を設けるのではなく、分解の各レベルの一つに対する三つの分離メモリ・ユニットとして設けることができる。このオプションは分離した読み取り・書き込みアドレス・ジェネレータを必要とする。

10

20

【0065】

デジタル画像を符号化する方法は、図17に示したような従来の汎用コンピュータ・システム 1700を使用して実行でき、例えば、図2、3および5に関する処理が、コンピュータ・システム 1700内に存在するアプリケーション・プログラムのようなソフトウェアとして備えることができる。特に、デジタル画像を符号化する方法中の工程は、コンピュータによって実行されるソフトウェア内の命令によって実行される。このソフトウェアは二つのパートに分割することができ、その一つはエンコーディング方法を実行し、他の一つは後者とユーザ間のユーザ・インターフェイスを管理する。ソフトウェアはコンピュータ読み取り可能媒体内に記憶することができ、例えば次に説明する記憶デバイスを含んでいる。このソフトウェアは、コンピュータ読み取り可能媒体からコンピュータにロードされ、コンピュータによって実行される。この種のソフトウェア、またはこれに記憶されたコンピュータ・プログラムを有するコンピュータ読み取り可能媒体は、コンピュータ・プログラム製品である。コンピュータ内のコンピュータ・プログラム製品を使用すると、本発明の実施の形態に基づくデジタル画像を符号化するための装置が好適に実行される。

30

【0066】

コンピュータ・システム 1700は、コンピュータ・モジュール 1701、キーボード 1702およびマウス 1703のような入力デバイス、プリンタ 1715および表示デバイス 1714を含む出力デバイスを備えている。モジュレータ-デモジュレータ(モデム)受信デバイス 1716が、例えば、電話回線 1721または他の機能媒体を介して接続可能な通信ネットワーク 1720間を相互連絡するコンピュータ・モジュール 1701によって使用される。モデム 1716はインターネット、および、ローカル・エリア・ネットワーク(LAN)またはワイド・エリア・ネットワーク(WAN)のような他のネットワーク・システムへのアクセスを得るのに使用することも可能である。

40

【0067】

コンピュータ・モジュール 1701は、一般的に、少なくとも一つのプロセッサ・ユニット 1705と、例えば、半導体ランダム・アクセス・メモリ(RAM)とリード・オンリ・メモリ(ROM)から形成されたメモリ・ユニット 1706と、ビデオ・インターフェイス 1707を含む入力/出力(I/O)インターフェイスと、キーボード 1702とマウス 1703およびオプションとしてジョイスティック(図示省略)のための I/O イン

50

ターフェイス 1713 と、モデム 1716 のためのインターフェイス 1708 とを備えている。記憶デバイス 1709 を備えており、一般的にハードディスク・ドライブ 1710 とフロッピー・ディスク 1711 を含んでいる。磁気テープ・ドライブ(図示省略)も使用できる。CD-ROM ドライブ 1712 が、一般的にデータの不揮発性ソースとして備えられている。コンピュータ・モジュール 1701 の要素 1705 から 1713 は、一般的に相互連結バス 1704 を介して、当該技術において知られているコンピュータ・システム 1700 の従来操作モードとなる方法で連絡されている。実施の形態を実行できるコンピュータの例としては、IBM-PC および互換性のあるサン・スパークステーションまたはこれから発展されたこれと同様のコンピュータ・システムが含まれる。

【0068】

10

一般的に、好適な実施の形態のアプリケーション・プログラムは、ハードディスク・ドライブ 1710 に存在し、プロセッサ 1705 によるその実行が読み取られるとともに制御される。プログラムの中間記憶およびネットワーク 1720 からくるどのデータも、多分ハードディスク・ドライブ 1710 と提携して半導体メモリ 1706 を使用して達成される。ある例において、アプリケーション・プログラムは、ユーザに提供され、CD-ROM またはフロッピー・ディスクに符号化され、対応するドライブ 1712 または 1711 を介して読み取られるか、或いは別 の方法として、モデム・デバイス 1716 を介してネットワーク 1720 からユーザによって読み取ることもできる。さらに、ソフトウェアも他のコンピュータ読取可能媒体からコンピュータ・システム 1700 にロードできる。この媒体には磁気テープ、ROM または集積回路、磁気・光学ディスク、コンピュータ・モジュール 1701 と他のデバイス間の無線または赤外線伝送チャネルと、PCMCI A カードのようなコンピュータ読取可能カード、およびインターネットと、E メール伝送とウェブサイトに記憶された情報を含むインターネット等々を含む。上述のものはコンピュータ読取可能媒体の単なる例である。他のコンピュータ読取可能媒体も、本発明の範囲と概念から外れることなく実行できる。

【0069】

20

デジタル画像を符号化する方法は、別 の方法としてデジタル画像を符号化する機能、又はサブ・ファンクションを実行する一つまたはそれ以上の集積回路のような専用ハードウェア内で実行できるのが好ましい。この種の専用ハードウェアにはグラフィック・プロセッサ、デジタル・プロセッサ、あるいは一つのまたはそれ以上のマイクロプロセッサおよび関連するメモリを含む。

[エントロピー符号化器(エンコーダ)]

30

デジタル画像のコード化表現を発生するためのエンコーダを次に説明する。

【0070】

画像の変換係数のコード化表現を生成するための方法を説明する。特に、この方法はクワドツリー(4 分木)を使用する埋め込みブロック・コーディングに関する。

【0071】

この方法は画像データのウエーブレット変換により最初に処理される。ウエーブレット処理の概観を添付図面を参照して次に説明する。

【0072】

40

図 18 を最初に参照して、元の画像 1801 が離散ウエーブレット変換(DWT)を使用して四つのサブ画像 1803 乃至 1806 に変換される。このサブ画像ないしサブバンドは、通常 LL1、HL1、LH1 および HH1 で示される。サブバンド名に付いている 1 の添え字はレベル 1 を示している。LL1 サブバンドは、元の画像のロウパス・デシメート・変形である。

【0073】

ここで使用されるウエーブレット変換は、例えばハール(Haar)基本ファンクション、Daub echies 基本ファンクション等で変形でき、またこれを含めることができる。LL1 サブバンドは次に使用され、第 2 離散ウエーブレット変換が図 19 に示したように適用され、サブバンド LL2(1908)、HL2(1909)、LH2(1910)、HH2(19

50

11) で示される。この処理は例えば図20に示したように続行され、ここにはLL4サブバンドが示されている。明らかに、分解の第四レベルは入力画像のサイズに依存して提供できる。最も低い周波数のサブバンドはDCサブバンドに該当される。図20の場合には、DCサブバンドはLL4サブバンドである。

【0074】

次に、各単一レベルDWTは、元の画像を得るように順次変換される。従って、J-レベルDWTは一連のJ単一レベルの逆DWTに逆変換できる。

【0075】

階層的に画像をコード化するために、DCサブバンドが最初にコード化される。こうして、残りのサブバンドがレベルを小さくする順にコード化される。これは4レベルDWTに対してであり、レベル4のサブバンドはDCサブバンド(LL4)の後にコード化される。すなわち、HL4、LH4およびHH4サブバンドである。レベル3(HL3、LH3およびHH3)のサブバンドが次にコード化され、これに続いてレベル2(HL2、LH2およびHH2)が、次にレベル1(HL1、LH1およびHH1)がコード化される。

【0076】

標準画像では、符号化化サブバンドは通常、画像中に「詳細」情報を含んでいる。量子化後のサブバンドは、しばしば値が希薄なアレイになり、実質的な圧縮がその希薄なマトリックス形態の有効なエンコーディングによって達成される。

【0077】

ここで図21を参照して、HH1のようなサブバンドのタイリングを示す。このサブバンドは、左上のコーナーから始まる 32×32 ブロックの係数を有するタイル2110、2120、2130、2140および2150であるのが好ましい。 32×32 なる用語は、それぞれ32の行(row)、 \times 32列(column)を意味している。

【0078】

以下の実施の形態の説明を進める前に、以後に使用する技術用語につき簡単に説明する。2進整数を表す数に対して、「ビットn」または「ビット数n」は、2進数nに該当し、例えば、8ビットの2進表現を仮定すると、10進数の9は、00001001として表される。この数において、ビット3は1に等しく、ビット2、1および0はそれぞれ0、0、1に等しい。さらに、画像の変換は、ビット・シーケンスによって表された各係数による横列と縦列に配列された係数を有するマトリックスとして表すことができる。結果的に、マトリックスについて話すと、三つのディメンジョン(次元)を有していると考えることができ、一番目のディメンジョンは横方向、二番目は縦方向、三番目はビットのシーケンス方向である。同じビット番号で各ビット・シーケンスを通っているこの三つのディメンジョン(3次元)空間内の平面は、「ビットプレーン」または「ビットプレーン」と呼ばれる。用語「ビットプレーン数n」は、ビット番号nを通るビットプレーンを意味する。

【0079】

説明を簡潔にするために、変換係数は、以後、固定小数点で符号のない2進整数フォーマットに、付加的な単一サイン・ビットを付して表す。すなわち、10進数の-9および9は、同じビット・シーケンス、すなわち、1001で表し、前者は負数を示す1に等しいサイン・ビットを有し、後者は正数を表す0に等しいサイン・ビットを有する。整数表示を使用することにおいて、係数は最も近い整数値に既に暗黙のうちに量子化されているが、これは必要ではない。さらに、圧縮の目的で、端数ビットに含まれるどの情報も通常無視される。

【0080】

画像フレームの域は、一連の隣接する画像係数を含んでいる。用語「係数」は、以後「画素」と交換可能に使用されるが、当該技術に習熟した人にとってよく理解される点では、前者が変換ドメイン(例えば、DWTドメイン)中で画素という意味で使用されるのが一般的である。これらのセットないし域Tは、(i, j)を係数座標としたとき、変換画像係数 $\{c_{i,j}\}$ を有するのもとして規定され、固定小数点でサインのない2進整数フォーマットに付加的な単一サイン・ビットを付して表わされる。

10

20

30

40

50

【0081】

現在ビットプレーンにおける画素のセット或いは域 T は、域中の各係数の M S B の数が現在のビットプレーンの値より小さい場合、重要ではないと言える。域の意味の概念を正確にするために、数学的規定が式(1)で与えられる。画素のセット或いは域 T は、次式であれば、ビットプレーン n に関して(またはこれにおいて)重要でないと言える。

$$|c_{i,j}| < 2^n, \text{ 全ての } c_{i,j} \in T \text{ に対して} \quad (1)$$

座標の集合類別 T により、次の部分集合 T のコレクション $\{T_m\}$ を定める。

【0082】

【数1】

$$T = \bigcup_{m=1}^n T_m, \quad T_n \cap T_m = \emptyset \quad \forall n \neq m$$

10

【0083】

換言すれば、一つの $c_{i,j} \in T$ 、従って $c_{i,j} \in T_m$ は、ただ一つの部分集合 T_m である。本件の T は正方形であり、集合 $\{T_m\}$ は、 T の4象限からなる集合(セット)である。

【0084】

提案された方法は、4分木を使用して埋め込み方法で一連の係数を符号化する。用語「埋め込み」の使用は、より低いビットプレーンでいずれかのビットをコード化する前に、より高いビットプレーン中の全てのビットをコード化することを意味している。次に、ビットプレーン6のいずれかのビットがコード化される前に、ビットプレーン7の各ビットがコード化され、順番に、ビットプレーン5のいずれかのビットがコード化される前に、ビットプレーン6の全ビットがコード化されるという順にコード化される。

20

【0085】

提案された方法の好適な実施の形態は、次の疑似コードを使用して実行される。この提案された方法は、係数の正方ブロックを符号化するのが好ましく、ブロックのサイズは2の累乗(一般的に 32×32 の係数)したものである。更に、この提案方法は4分木類別を使用している。すなわち、各セット(集合)或いは域は、その四つの象限に類別され、これによって、常時2の累乗に等しいディメンジョンで正方形が維持される。開始時に、提案方法は三つのリストを初期化する。すなわち、重要でない域(有効数字でない部分)(LIR)のリストと；重要でない係数のリスト(LIC)と；重要な(有効数字部分)係数のリスト(LSC)である。単一の係数が、係数の重要性に依存して、重要でないセットのリスト(LIR)から除去されると、これらのセットは、重要でない係数のリスト(LIC)か、重要な係数のリスト(LSC)のいずれかに付加される。

30

【0086】

この提案方法は次のように初期化される。LICおよびLSCは空になるように初期化される。LIRは入力ブロックの4象限を包含するセット(集合)である。この方法は n_{max} を見つけて、コード化することによって開始され、ここで n_{max} はセット内の全ての係数に対して1ビットを含む最大のビットプレーンである。従って、この提案方法は次のようにして進行される：

40

1. $n = n_{max}$ に設定

2. 重要でない係数のリスト(LIC)中の各係数に対し、

- ・ 係数のコード・ビット n (すなわち、その重要度(有効数字))
- ・ ビットが1(即ち、重要である場合)、サイン・ビットにコード化される。係数をLSCの最後に付加され、LICから係数が除去される。

【0087】

3. 重要でない域のリスト(LIR)中の各域 T に対し、

- ・ 重要な T (シグニフィカンス)をコード化。

【0088】

- ・ T が重要(有効数)であり、一つより多い係数から構成され、従って、 T がその4象

50

限に仕切られ、それらが L I R の最後に付加される。T がリストから除去される。

【0089】

- ・ T が単一係数であれば、
- ・ T が L I R から除去される。

【0090】

- ・ T が有効数であれば、サイン・ビットがコード化され、T が L S C の最後に付加される。

【0091】

- ・ 他に、T を L I C のリストの最後に付加する。

【0092】

4. 重要な係数 L S C (ステップ 3 中のリストに付加されたものを除く) のリスト中の各係数 $c_{i,j}$ に対し、

- ・ $c_{i,j}$ のビット n をコード化、

5. n を小さくして、ステップ 2 に進む。

【0093】

上述のことから、出力ビットストリームは普通、次の形態をとる、

... L I C ' L I R ' L S C '

ここで、L I R ' はステップ 3 で実行されたコード化表現であり；L I C ' はステップ 2 で実行されたコード化表現であり；L S C ' はステップ 4 で実行されたコード化表現である。しかし、エンコーディング工程の最初の繰り返し中、L I C と L S C は空であり、従って、最初の繰り返しの出力ビットストリームは L I R ' の形態をとることに注意しなければならない。

【0094】

この提案した方法に加えて、L I C および L S C をコード化するときに、簡単なHuffman コード（またはより良いのはGolombコード）をビットのグループ（例えば 4 ビットのグループ）がコード化するのに使用される。さらに、域の各象限の有効数をコード化するときに、15 レベルのHuffman コードが 4 象限（1 象限は重要でなければならず、従って、重要なパターンは 15（16 ではない）の内の一つとなる）のそれぞれの異なるパターンの重要なパターンを示すのに使用できる。エントロピー符号化の他の形態として、例えば 2 進演算コーディングなど使用して、残りの冗長性を利用する。

【0095】

別の実施の形態として、T が 2×2 ブロックの係数からなる場合、ステップ 3 における提案された方法は、次のサブステップを形成する。 2×2 ブロックの各係数の重要度（意味）をコード化して出力すると直ちに、これらが重要であれば、対応するサインが出力されるとともに、その係数が適切に L S C または L I C の最後に付加される。後者のサブステップにおいて、重要でない係数が L I C リストに付加される。

【0096】

好みしくは、この提案した方法は 32×32 ブロックのデータ・係数を符号化する。説明の目的のみで、 4×4 ブロックの係数の次なる例が、この提案した方法に基づいて符号化される。

【0097】

【数 2】

$$\begin{bmatrix} 31 & 16 & 0 & 0 \\ 15 & 17 & 0 & 0 \\ 9 & 7 & 1 & 0 \\ 5 & 3 & 1 & 0 \end{bmatrix}$$

【0098】

10

20

30

40

50

上述のブロックは四つの象限 A、B、C および D からなる。シンボル A は、ブロックの左上 (2 × 2) 象限、B は右上、C は左下、また、D は右下の象限を表す。さらに、シンボル A 1 はそれぞれ A の左上の画素、A 2 は右上、A 3 は左下、A 4 は右下の画素を示している。同様にして、B 1 は B の左上、というように画素の残りの部分にも適用される。

【0099】

この提案方法によれば、 n_{max} が最初に決定されるが、この場合 4 である。すなわち、各係数の最も重要なビットはビットプレーン 4 またはそれ以下にある。注意しなければならないのは、ビットプレーンの番号付けは 0 から始まることである。変数 n_{max} は 4 ビットでコード化される（係数が圧迫されているので、 n_{max} は 0 から 15 の間となる）。最初に、

$LIC = \emptyset$, $LIR = \{A, B, C, D\}$ および $LSC = \emptyset$ であり、ここにシンボル \emptyset は、空きリストを示すのに使用される。

【0100】

従って、この提案方法によれば、ビットプレーンは繰り返してコード化される。この処理はビットプレーン $n = n_{max} = 4$ で開始され、減少数 n は各繰り返しにつき 1 ずつである。

1. $n = n_{max} = 4$ につき、

- 最初に、リスト LIC 中の各係数がコード化される。何もないで、コード化は実行されない。
- 次に、リスト LIR 中の各域の重要度がコード化される。

【0101】

・域 A に対して、ビットプレーン $n = 4$ において重要であるから、1 ビットが出力される。次に、A の 4 象限が、すなわち、A 1、A 2、A 3 および A 4 がリスト LIR の最後に付加され、A が除去される。ここで $LIR = \{B, C, D, A 1, A 2, A 3, A 4\}$ である。

【0102】

- 域 B に対して、ビットプレーン $n = 4$ において重要でないから、0 ビットが出力される。

【0103】

- 域 C に対して、0 ビットが出力される。

【0104】

- 域 D に対して、0 ビットが出力される。

【0105】

・域 A 1 に対して、1 ビットが出力される。A 1 が単一係数 31 からなるので、A 1 が LIR から除去される。31 (または A 1) が重要であるから、これが LSC に (またはそのロケーションがブロックに) 付加される。A 1 のサイン・ビットは (0) も出力される。

【0106】

・域 A 2 に対して、1 ビットが出力される。A 2 が単一の重要な係数 16 からなるので、 LIR から除去され、 LSC の最後に付加される。A 2 のサイン・ビット (0) も出力される。ここで $LSC = \{31, 16\}$ である。

【0107】

・域 A 3 に対して、0 ビットが出力される。単一の重要でない係数であるから、 LIR から除去され、係数 15 が LIC に付加される。ここに、 $LIC = \{15\}$ である。

【0108】

・域 A 4 に対して、1 ビットが出力される。A 4 が単一の重要な係数 17 からなるので、 LIR から除去され、 LSC の最後に付加される。A 4 のサイン・ビット (0) も出力される。ここで、 $LSC = \{31, 16, 17\}$ である。

・最終ステップにおいて付加されなかった LSC 中の各係数がここでコード化される。何もないで、コード化は実行されない。

10

20

30

40

50

【0109】

従って、第1繰り返しにおいて、この提案した方法が次のビットストリームを出力する。

1 0 0 0 1 0 1 0 0 1 0

この段階において、ビットプレーン4（及びこれより高い）中の全てのビットがコード化される。符号化されたビットストリームからビットを読み出すことにより、復号器がビットプレーン4（及びより高い）を再構築できる。このデコーディング方法は、重要度の判定がビットストリーム（これが、なぜ重要度の決定がビットストリームに書き込まれるかの理由である）からの読み取りによって決定されることを除いて、同じである。他の係数・ビットがそのままで簡単に読み取られる。デコーダ実行パスは、エンコーダのものと同様であり、これによってデコーダは、読み取られた新しい各ビットの意味がわかっていることに注意しなければならない。

2. $n = 3$ につき、

最初に、 $LIC = \{15\}$ 、 $LIR = \{B, C, D\}$ および $LSC = \{31, 16, 17\}$ である。

- 第1に、 LIC 中の各係数のビット $n = 3$ をコード化する。すなわち、1ビットが係数 15 と、サイン・ビット(0)に対して出力される。これは重要であるので(1ビットが出力される)、サイン・ビットが出力され、係数 15 が LIC から除去され、 LSC の最後に付加される。これによって、 $LSC = \{31, 16, 17, 15\}$ となる。

- LIR 中の域の各々の重要度がここでコード化される。

【0110】

- 域 B に対して、0ビットが出力される。

【0111】

- 域 C に対して、ビットプレーン $n = 3$ において重要であるから、1ビットが出力される。域 C が4象限 $C00, C01, C10$ および $C11$ に仕切られ、 LIR の最後に付加される。 C が LIR から除去される。従って、ここに $LIR = \{B, D, C00, C01, C10, C11\}$ である。

【0112】

- 域 D に対して、0ビットが出力される。

【0113】

- 位置 $C00$ に対して、1ビットが出力される。 $C00$ が単一の重要な係数 9 からなるので、 LIR から除去され、 LSC の最後に付加される。 $C00$ のサイン・ビット(0)も出力される。ここに $LSC = \{31, 16, 17, 15, 9\}$ を得る。

【0114】

- 域 $C01$ に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数 7 を LIC に付加する。ここで $LIC = \{7\}$ である。

【0115】

- 域 $C10$ に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数 5 を LIC に付加する。ここに $LIC = \{7, 5\}$ である。

【0116】

- 域 $C11$ に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数 3 を LIC に付加する。ここに $LIC = \{7, 5, 3\}$ である。

- ここで、 LSC の各係数のビット $n = 3$ をコード化する（上記に単に付加するだけではない）。

【0117】

- 31、16 および 17 のビット $n = 3$ として、それぞれ 1、0 および 0 を出力する。

【0118】

従って、二回目の繰り返しにおいて、この提案した方法は次のビットストリームを出力する。

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

10

20

30

40

50

3. $n = 2$ につき、

最初に、 $LIC = \{ 7, 5, 3 \}$ 、 $LIR = \{ C, D \}$ および $LSC = \{ 31, 16, 17, 15, 9 \}$ を得る。

- 第1に、 LIC 中の各係数のビット $n = 2$ (またはビットプレーン $n = 2$ における重要度と等価である) をコード化する。すなわち、7、5 および 3 それぞれに対して 1、1 および 0 を出力する。加うるに、7(0) および 5(0) のためのサイン・ビットが出力され、これらの係数が LSC に移動される。 LIC 中に 3 が残る。

- 次に、 LIR 中の各域の有効数が符号化される。

【0119】

- 域 B に対して、0 ビットが出力され、また、域 D に対して 0 ビットが出力される。

10

- 最後に、 LSC 中の係数の各々に対するビット $n = 2$ を更新する (上記に付加せず)。

【0120】

- 31、16、17、15 および 9 に対して、それぞれ 1、0、0、1 および 0 を出力する。

【0121】

従って、三回目の繰り返しにおいて、この提案した方法は次のビットストリームを出力する。

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

ビットプレーン 0、またはいくつかの他の有限点まで、この様態を続行する。特定の有限コードを使用した場合、(三つの)サブパスのどれか一つの後、終了することができることに注意しなければならない。(基本的に、FF は有限コードとしてリザーブされ、有限コードを慎重に挿入しなければ、コード化ビットストリームを決して強制的に FF に含ませることはない)。

20

【0122】

これまでに説明したように、この方法は係数の 32×32 ブロックをエンコーディングするのに好適に利用できる。これらの状況において、元の象限 A、B、C、D は各々 16×16 の係数からなり、また域 A1、A2、...、D4 は各々 8×8 の係数からなる。従って、 32×32 ブロックの符号化において、ブロックは 4 分木方法に基づいて 5 回仕切られるが、一方、与えた例においては 4×4 ブロックが 2 回のみで仕切られる。

30

【0123】

デコーディング工程は、エンコーディング工程を単に模倣してコード化された表現から画素を再構築する。

【0124】

図 2.2 を参照して、この提案方法を実行するための第1の好ましい実施の形態に基づくエンコーダを説明する。係数・エンコーダ 2200 は、対応するデータ 2202 を得て、出力符号化データ 2204 の連続した流れを提供するように設計されている。このエンコーダ 2200 は、次のロジック部を含んでいる。すなわち、ビットプレーン・コンバータ 2206；ビットプレーン・ツリー・ビルダー 2208-0 から 2208-15；ビットプレーン・エンコーダ 2210 である。

40

【0125】

エンコーダ 2200 は、ビットプレーンで 32×32 の係数を記憶するためのメモリ 2214 と；係数のサイン・ビットを記憶するためのメモリ 2216 と；ビットプレーン 0 から 15 を記憶するためのメモリ 2218-15 から 2218-0 も含んでいる。エンコーダ 2200 は、域の第1リストを記憶するためのメモリ 2226 と；域の第2リストを記憶するための FIFO メモリ 2228 と； LSC 値のリストを記憶するためのメモリ 2222 と； LIC 値のリストを記憶するためのメモリ 2224 をさらに含んでいる。

【0126】

エンコーダ 2220 は、次のようにして動作する。最初に、 32×32 入力係数・データがラスター順にメモリ 2212 に記憶される。ビットプレーン・コンバータ 2206 は、入

50

力係数・データを読み取り、このデータを、ビットプレーン0からビットプレーン15までの16のビットプレーンに変換し、このデータは続いてメモリ2214に記憶される。ビットプレーン・コンバータ2206はまた 32×32 ブロックの係数のサイン・ビットも判定し、このサイン・ビットがメモリ2216に記憶される。

【0127】

ビットプレーン・ツリー・ビルダー2208-0から2208-15が各々 32×32 係数・データを読み取り、域に対応するノードと 1×1 の画素を有する4分木構造を構築する。各ツリーにおいて、ノードは、域またはこのノードに対応する画素がビットプレーンのために重要であれば、1にセットされる。そのノードに対応する域又は画素のための重要なビットが、ビットプレーンより大きいか、又は小さければ、ノードはゼロにセットされる。

【0128】

次に図23を参照して、 32×32 ブロックのためのビットプレーンnにおけるビットプレーン・ツリービルダー2208-nによって構成された構築ツリー2300を説明する。簡潔にする目的で、ツリーの一部のみを示す。このツリー2300は、 1×1 画素レベルまで下げるブロックの各4分木の仕切りを表すノードを含んでいる。この場合において、全 32×32 ブロックがシンボル0によって表される。 16×16 の左上象限がノードAで表され、 16×16 右上象限がBで表され、 16×16 の左下象限がノードCで表され、 16×16 の右下象限がノードDで表される。ノードA1、A2、A3およびA4が、4分木のAの左上、右上、左下および右下の 8×8 の4分木の仕切りをそれぞれ表している。同様にして、ノードA11、A12、A13およびA14は、象限A1の4分木の左上、右上、左下および右下の 4×4 4分木の仕切りをそれぞれ表している。同様にして、ノードA111、A112、A113およびA114は、象限A11の4分木の左上、右上、左下および右下の 2×2 の4分木の仕切りをそれぞれ表している。最後に、ノードA1111、A11112、A11113およびA1114は、象限A111の4分木の左上、右上、左下および右下の 1×1 4分木の仕切りをそれぞれ表している。ツリーの残りの部分(図示省略)は 32×32 ブロックの各 1×1 象限まで下がって同様の方法で表される。

【0129】

ビットプレーン・ツリー・ビルダー2208-nが、象限順(例えば、A1111、A1112、A1113およびA1114)で係数を読み取ることによってビットプレーンnを逆からこの種のツリーを構築する。そのノードに対応する域又は画素のための重要なビットが、ビットプレーンより大きいか、又は小さければ、ノードはツリービルダーによってゼロにセットされる。こうして、ビットプレーン・ツリー・ビルダーは、次のフォーマットで各ノードのための重要な情報を出力する。

【0130】

A B C D A1 A2 A3 A4 A11 A12 A13 A14 A21, ..., D44 A111, ..., D444, ..., A1111, D4444

ビットプレーン・ツリー・ビルダー2208-0から2208-15の各々からの出力は、それぞれのビットプレーン・ツリー・メモリ2218-0から2218-15に記憶される。

【0131】

ビットプレーン・エンコーダ2210が、ビットプレーン・ツリー・メモリの各々を読み取り、次にビットプレーン・ツリー・メモリ2218-15で開始する。ビットプレーン・エンコーダ2210が、ノードA、B、CおよびDに対応するビットプレーン・メモリ2218-15内に記憶された四つの重要なビットA、B、CおよびDを次に読み取ることによって処理が開始される。ビットプレーン・エンコーダは、メモリ2226内の域の第1リスト中のこれらのノード{A、B、C、D}のリストを記憶する。ビットプレーン・エンコーダ2210が次の操作に進行する:

1. ビットプレーン・エンコーダ2210が、第1域リスト上の第1ノード(域)に対

10

20

30

40

50

応するビットプレーン・ツリー内のビットを読み取る。

【0132】

a. ビットが重要であれば、エンコーダが2進の1を出力する。エンコーダが FIFO2228上の第2域リスト内のノードのチルドレン(子供)を記憶し、第1域リスト2226からのノードを除去する。

【0133】

b. ビットが重要でない場合、エンコーダが2進の0を出力し、第1域リスト2226上のノードを維持する。

【0134】

2. ビットプレーン・エンコーダ2210が、第1域リスト内の全てのノードが読み取られるまで、ステップ1を繰り返す。 10

【0135】

3. ビットプレーン・エンコーダが、FIFO2228上の第2域リスト内の第1ノードに対応するビットプレーン・ツリー中のビットを読み取る。

【0136】

a. ビットが重要であり、そのノードに関する子供がある(すなわち、ツリー内のそのノードの直下にノードがある)場合、エンコーダが2進のゼロを出力し、このノードを第1域リスト2226上のノードに入れる。

【0137】

b. ビットが重要であり、そのノードに関する子供がなければ、エンコーダが2進のゼロを出力し、このノードが対応する画素に対するインデックスとしてLICリスト2224に記憶される。 20

【0138】

c. ビットが重要であり、そのノードに関する子供がなければ、エンコーダが2進の1と対応するサイン・ビットを出力し、このノードが対応する画素に対するインデックスとしてLICリスト2222に記憶される。

【0139】

d. ビットが重要であり、そのノードに関する子供があれば、エンコーダが2進の1を出力し、第2域リストからノードを除去する。さらに、ノードの子供を第2域リストに付加する。 30

【0140】

e. 第2域が空きであれば、エンコーディング工程がそのビットプレーン・ツリーを完成させる。

【0141】

ビットプレーン・エンコーダが残りのビットプレーン・ツリーに対して次にこの操作を繰り返す。現在のビットプレーン・ツリーへの操作の開始時における域の第1リストには、前回のビットプレーン・ツリーへの前回操作から残っている域が含まれている。これらの出力は出力ストリームのLIR部分に対応する。現在のビットプレーン・ツリーに対する現在の操作の完了後、ビットプレーン・エンコーダがLICとLSCビットを符号化し、出力する。 40

【0142】

ビットプレーン・エンコーダが、リスト上の第1画素に対するインデックスのためのLICリストを読み取り、ビットプレーン・メモリ2214から必要とされるビットを選択する現在ビットプレーン数を使用することによってLICビットを符号化する。選択されたビットが2進の0であれば、エンコーダがゼロを出力する。選択されたビットが2進の1であれば、エンコーダが2進の1を、メモリ2216からの画素のサイン・ビットとともに出力する。こうしてエンコーダはLICからインデックスを除去し、これをLSCリストに付加する。一度、インデックスがリストから除去されると、残りのインデックスは再編成されるのが好ましい。エンコーディング(符号化)は、一度、LICが通過すると、完了する。 50

【0143】

ビットプレーン・エンコーダが、リスト上の第1画素に対するインデックスのためのLICリストを読み取り、ビットプレーン・メモリ2214から必要とされるビットを選択する現在ビットプレーン数を使用することによってLICビットを符号化する。こうして、選択されたビットが出力される。ビット・プレン・エンコーダは、読み取られるべきLSCリスト中の画素数を示す長さ値を記憶するためのカウンタも含んでいる。LSC符号化の終了後、この値が更新され、LIRとLICからの新しい要素を付加することができる。

【0144】

この提案方法のエンコーディング（符号化）とデコーディング（復号）工程は、図24に示したような従来の汎用コンピュータを使用して実行されるのが好ましく、その工程がこのコンピュータを実行するソフトウェアとして備えることができる。特に、コーディング及び（又は）デコーディング方法のステップは、コンピュータによって実行されるソフトウェア内の命令によって実行される。ソフトウェアは二つの別々のパートに分かれており、その一つがデコーディング及び（又は）コーディング方法を実行し、また他方の一つが後者とユーザ間のユーザ・インターフェイスを制御する。ソフトウェアは、例えば次に説明する記憶デバイスを含むコンピュータにより読み取可能な記憶媒体内に記憶できる。このソフトウェアはコンピュータ読み取可能な記憶媒体からコンピュータ内にロードされ、コンピュータによって実行される。この種のソフトウェア、又はこれに記録されたコンピュータ・プログラムを有するコンピュータにより読み取可能な記憶媒体は、コンピュータ・プログラム製品である。コンピュータ内でこのコンピュータ・プログラム製品の使用は、本発明によるデジタル画像をエンコーディング（符号化）し、また、デジタル画像のコード化された表現をデコーディングするための有利な装置を好適に実行する。

10

20

30

【0145】

コンピュータ・システム2400は、コンピュータ2401と、ビデオ表示装置2416と、入力デバイス2402、2403を備えている。加えて、コンピュータ・システム2400は、通信インターフェイス2421を介して、一つまたはそれ以上の他のコンピュータに、モデム通信バス、コンピュータ・ネットワーク等々のような適切な通信チャネル2422を使用して接続することができる。コンピュータ・ネットワークはローカル・エリア・ネットワーク（LAN）、ワイド・エリア・ネットワーク（WAN）、インターネットおよび（または）インターネットを含めることもできる。

30

【0146】

コンピュータ2401自体は、中央処理装置（以後単にプロセッサという）2405と、ランダム・アクセス・メモリ（RAM）とリード・オンリー・メモリ（ROM）とを含むメモリ2406と、入力/出力（I/O）インターフェイス2408と、ビデオ・インターフェイス2407と、図24のブロック2409で概略的に示された一つまたはそれ以上の記憶デバイスとを備えている。記憶デバイス2409は、次の内の一つまたはそれ以上を含んでいる。すなわち、フロッピー・ディスク、ハードディスク・ドライブ、磁気・光学ディスク・ドライブ、CD-ROM、磁気テープまたは当該技術において周知の他の多数の不揮発性記憶デバイスである。要素2405から2409、2421はそれぞれ一般的に、データ、アドレスおよび制御バスからなるバス2414を介して一つ、またはそれ以上の他のデバイスに接続される。

40

【0147】

ビデオ・インターフェイス2407がビデオ表示装置2416に接続され、コンピュータ2401からのビデオ信号を提供してこのビデオ表示装置2416に表示する。コンピュータ2401を作動させるユーザ入力は、一つ又はそれ以上の入力デバイス2402、2403によって提供できる。例えば、オペレータはキーボード2402および（または）マウス2403のような指示デバイスを使用することができる。

【0148】

システム2400は図示の目的で簡潔に示されているが、他の形態も使用できる。実施の

50

形態を実行できる例としてのコンピュータには、IBM-PC/ATまたは互換機を含み、その一つにPCのマッキントッシュ（商標名）ファミリ、サン・スパークステーション（商標名）等々がある。上述したものは実施の形態を実行できるコンピュータのタイプの単なる一例である。一般的には、以後に説明する実施の形態の処理は、プロセッサ2405を使用する読み取りおよび制御されたコンピュータにより読み取可能な記憶媒体としてハードディスク・ドライブ（図24において通常、ブロック2409で示される）に記憶されたソフトウェアまたはプログラムとして内在している。プログラムと画素データおよびネットワークから取り入れられたデータの中間記憶は、半導体メモリ2406を使用して、ハード・ディスク・ドライブ2409と提携して達成することができる。

【0149】

10

ある例において、プログラムは、CD-ROMまたはフロッピー・ディスク（両者ともブロック2423で概略的に示す）上に符号化されユーザに提供されるか、または別の方針として例えば、コンピュータに接続されたモデム・デバイスを介してネットワークからユーザが読み取ることができる。さらに、ソフトウェアは、他のコンピュータ読み取可能媒体からコンピュータ・システム2400にロードすることもできる。この媒体には磁気テープ、ROMまたは集積回路、磁気・光学ディスク、コンピュータと他のデバイス間の無線または赤外線伝送チャネル、PCMCI Aカードのようなコンピュータ読み取可能カード、およびEメール通信を含むインターネットおよびイントラネットと、ウェブサイト等に記憶された情報を含む。上述したものは適切なコンピュータ読み取可能媒体の単なる例である。他のコンピュータ読み取可能媒体も実行可能である。

20

【0150】

エントロピー符号化器から転じて符号化方法及び装置に説明を戻したが、逆に、符号化された画像を復号により再生しても良い。この場合、画像はまず最初に完全にエントロピー復号され、その復号された画像に、前述の符号化方法の略逆の処理が適用される。

【0151】

上述の説明から明白なように、本発明の実施の形態はコンピュータおよび画像/データ処理産業に適用可能である。

【0152】

30

これまでに説明したものは、本発明のある実施の形態のみで、修正および（または）変更も本発明の範囲と概念から外れることなく可能であり、実施の形態は例であって、限定ではない。

【0153】

【発明の効果】

以上説明したように本発明によれば、例えば少ないバッファの容量で、複数レベルのDWTを効率良く行うことができるという効果がある。

【図面の簡単な説明】

【図1】画像の4レベルの離散ウェーブレット変換の種々のサブバンドを示す図である。

【図2】本発明の好適な実施の形態の変形例としての画像の水平タイリングを示す図である。

【図3】本発明の好適な実施の形態に係る画像の垂直タイリングを示す図である。

40

【図4】本発明の好適な実施の形態の符号化器を示す図である。

【図5】図3のタイリングと好適なタイル走査順を示す図である。

【図6】図4の圧縮エンジンをより詳しく示す図である。

【図7】本発明の好適な実施の形態に係るリオーダ・バッファを示す図である。

【図8】図7のリオーダ・バッファの充填順を示す図である。

【図9】本発明の好適な実施の形態に係る垂直フィルタのハードウェア構成を示す図である。

【図10】本発明の好適な実施の形態に係る水平フィルタのハードウェア構成を示す図である。

【図11】本発明の好適な実施の形態に係るサブバンド・バッファのハードウェア構成を

50

示す図である。

【図12】図11のサブバンド・バッファの充填順および走査順を示す図である。

【図13】本発明の好適な実施の形態に係る制御状態マシンのハードウェア構成を示す図である。

【図14】本発明の好適な実施の形態に係る部分バンド・バッファのハードウェア構成を示す図である。

【図15】図14の部分バンド・バッファの充填順および走査順を示す図である。

【図16】図6の制御状態マシン・モジュールによる処理の流れを示すフローチャートである。

【図17】本発明の好適な実施の形態を実行する汎用コンピュータの概略ブロック図である。 10

【図18】元の画像とそのDWT変換を示す図である。

【図19】図18の元の画像の第2レベルDWT変換を示す図である。

【図20】図18の元になる画像の4レベルDWT変換を示す図である。

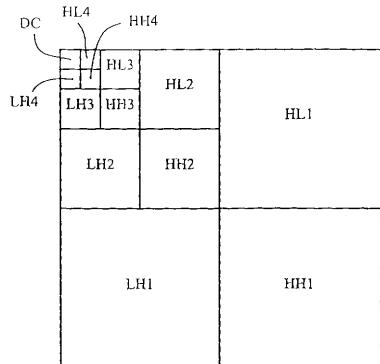
【図21】 32×32 ブロックにサブバンドをタイリングする図である。

【図22】符号化器の実施の形態を示す図である。

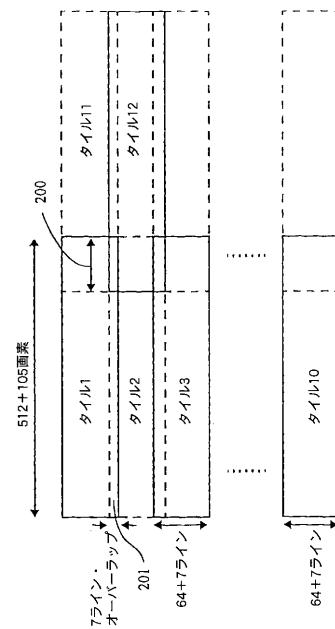
【図23】図22のビットプレーン・ツリー・ビルダーによって構築されたツリーの一部を示す図である。 20

【図24】説明した符号化方法を実行するための汎用コンピュータの構成を示すブロック図である。

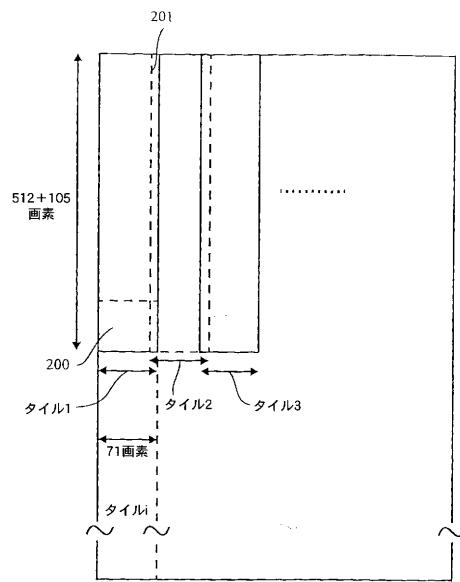
【図1】



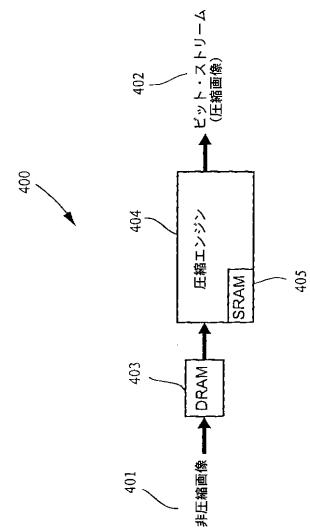
【図2】



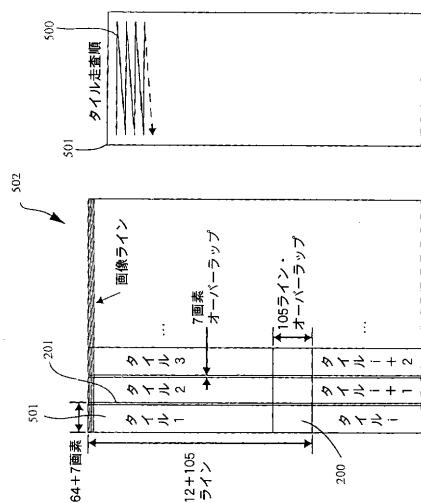
【図3】



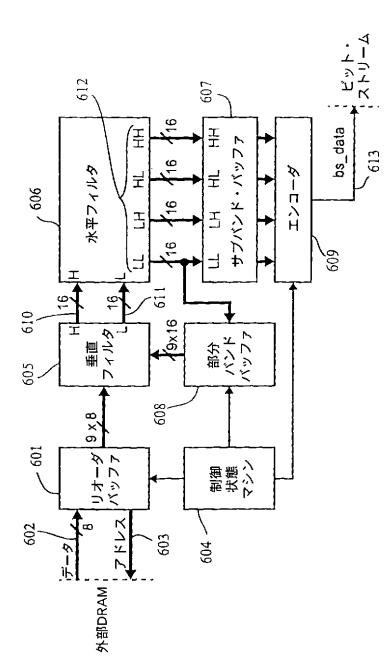
【図4】



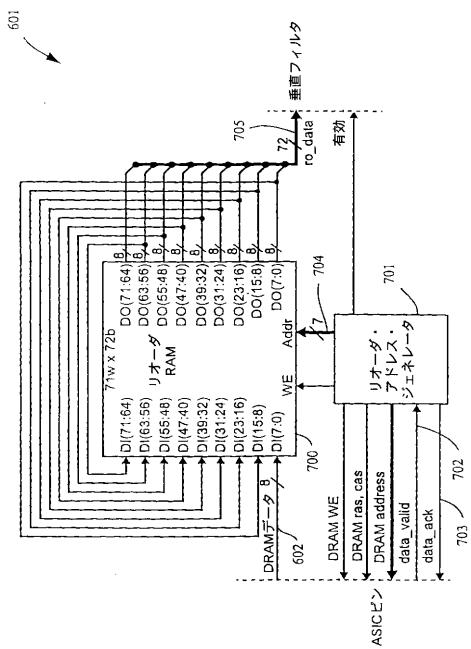
【図5】



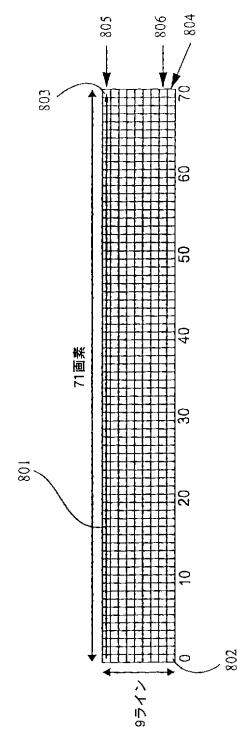
【図6】



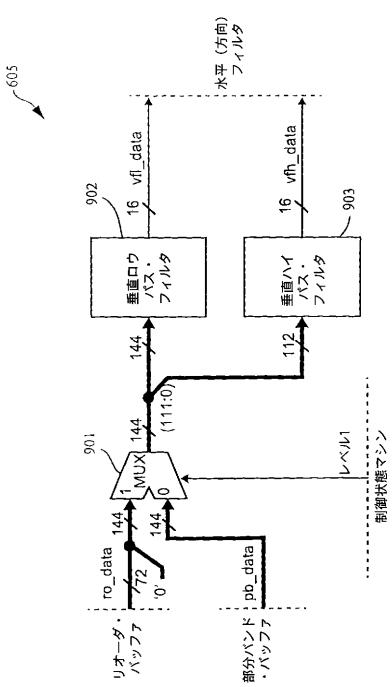
【図7】



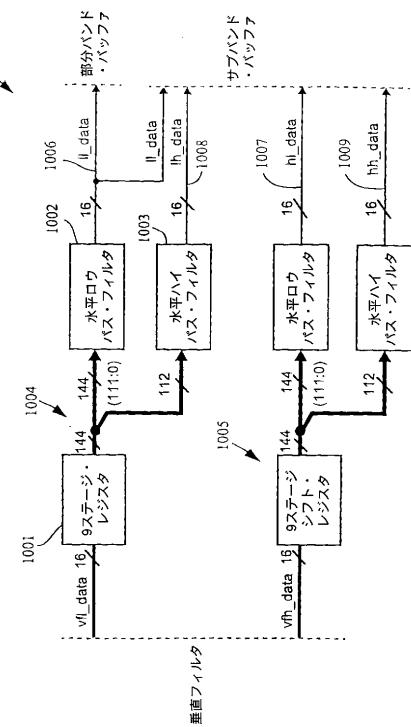
【図8】



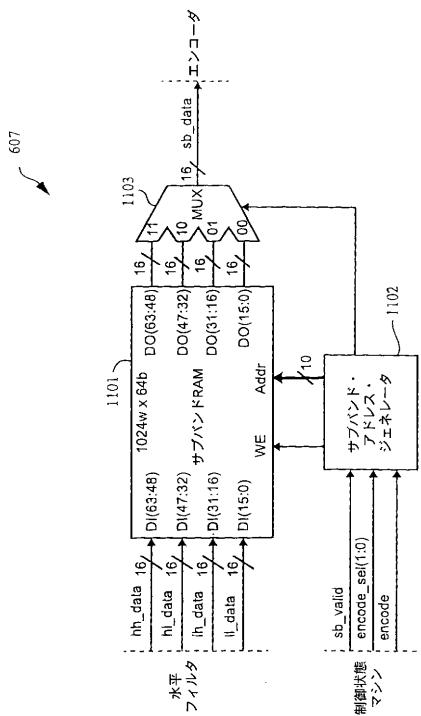
【図9】



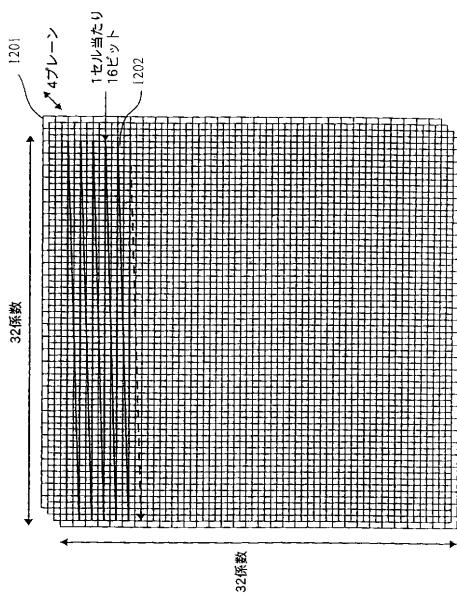
【図10】



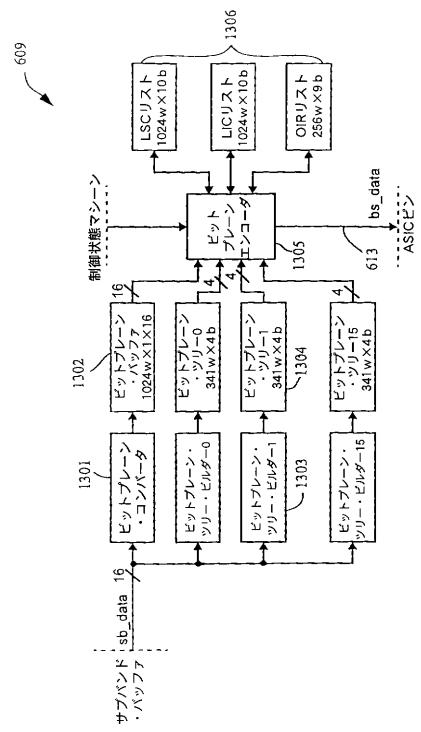
【図11】



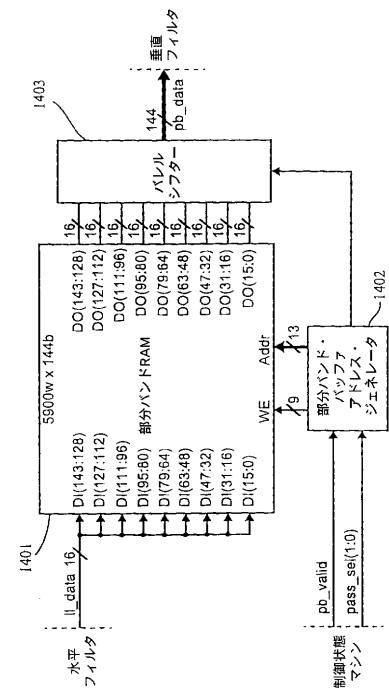
【 図 1 2 】



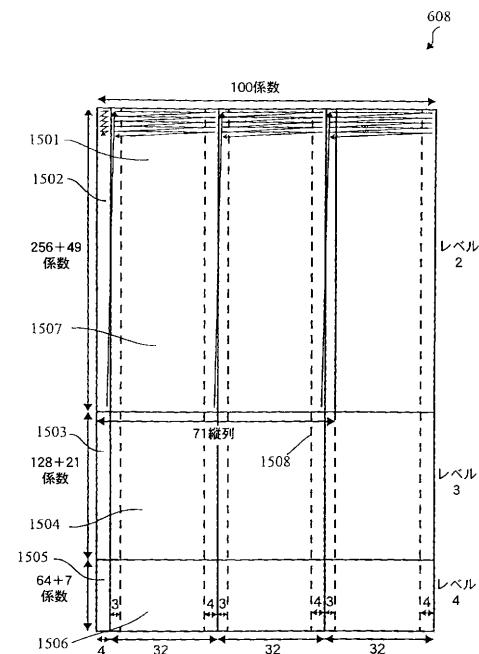
【図13】



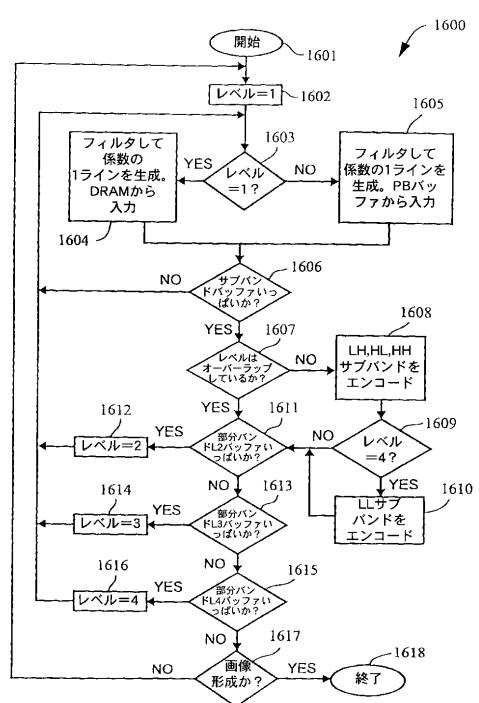
【 図 1 4 】



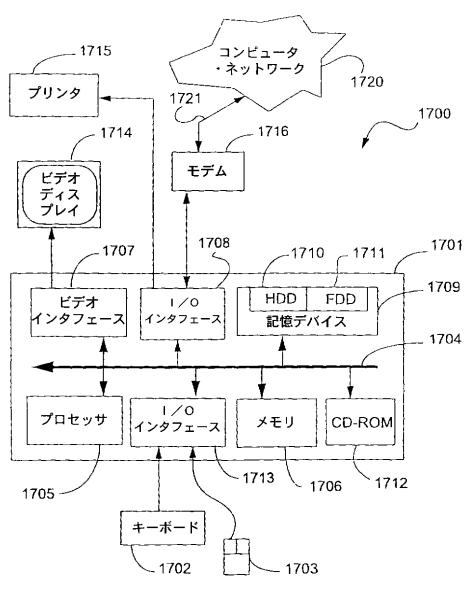
【図15】



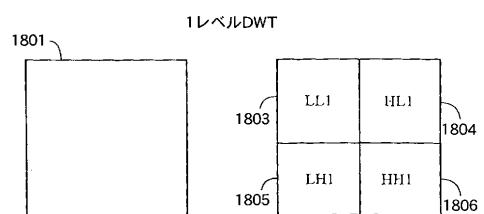
【図16】



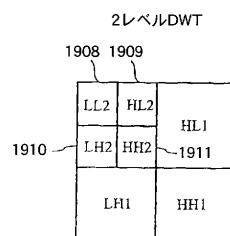
【図17】



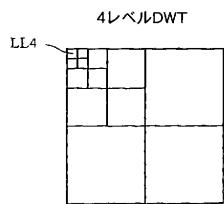
【図18】



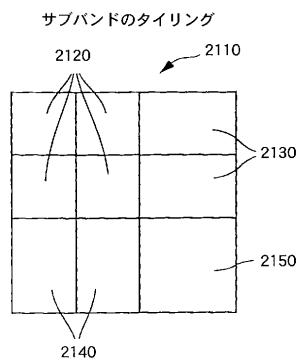
【図19】



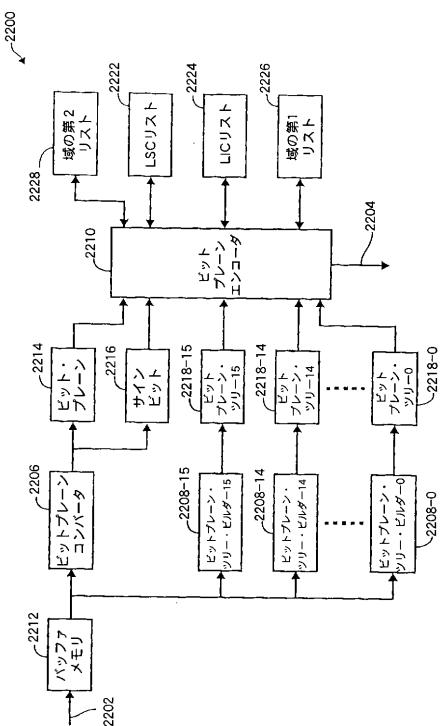
【図20】



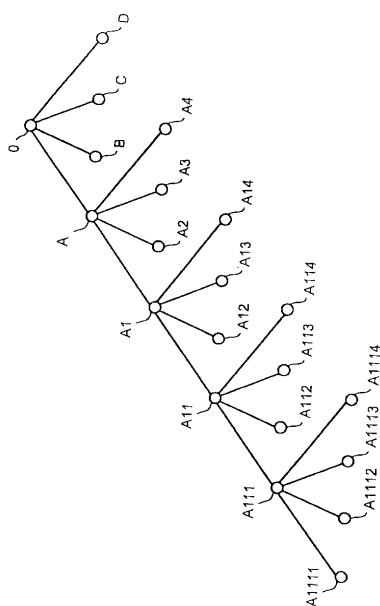
【図21】



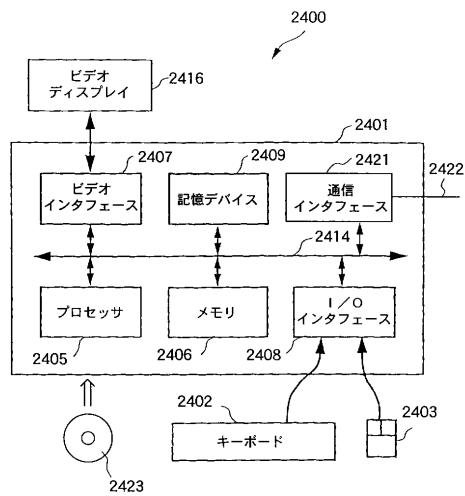
【図22】



【図23】



【図24】



フロントページの続き

(72)発明者 ポール レイモンド ヒギンボトム

オーストラリア国 2113 ニュー サウス ウエールズ州, ノース ライド, トーマス ホルト ドライブ 1 キヤノン インフォメーション システムズ リサーチ オーストラリア プロプライエタリー リミテッド内

(72)発明者 エリック マヤーニ

フランス国 レンヌ - アタラント, セデックス セッソン - セヴィニエ 35517, リュ ドゥラ トゥッシュ - ランペール キヤノン リサーチ センター フランス エス. エー. 内

(72)発明者 ジェームス フィリップ アンドリュー

オーストラリア国 2113 ニュー サウス ウエールズ州, ノース ライド, トーマス ホルト ドライブ 1 キヤノン インフォメーション システムズ リサーチ オーストラリア プロプライエタリー リミテッド内

審査官 堀井 啓明

(56)参考文献 特開平11-004343 (JP, A)

特開平10-336659 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N1/41-1/419

H04N7/12-7/137