

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5966347号
(P5966347)

(45) 発行日 平成28年8月10日 (2016. 8. 10)

(24) 登録日 平成28年7月15日 (2016. 7. 15)

(51) Int. Cl.		F I			
HO4N	1/41	(2006.01)	HO4N	1/41	B
HO4N	19/93	(2014.01)	HO4N	19/93	
HO3M	7/40	(2006.01)	HO3M	7/40	

請求項の数 15 (全 65 頁)

(21) 出願番号	特願2011-280579 (P2011-280579)	(73) 特許権者	000002185
(22) 出願日	平成23年12月21日 (2011.12.21)		ソニー株式会社
(65) 公開番号	特開2013-131938 (P2013-131938A)		東京都港区港南1丁目7番1号
(43) 公開日	平成25年7月4日 (2013.7.4)	(74) 代理人	100082131
審査請求日	平成26年12月19日 (2014.12.19)		弁理士 稲本 義雄
		(74) 代理人	100121131
			弁理士 西川 孝
		(72) 発明者	保坂 和寿
			東京都港区港南1丁目7番1号 ソニー株式会社内
		審査官	松永 隆志

最終頁に続く

(54) 【発明の名称】 画像処理装置および方法

(57) 【特許請求の範囲】

【請求項1】

画像データから生成された複数の係数データを所定の数毎に1組とし、毎周期1組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化する有効桁数符号化部と、

値が0の係数データのみで構成される組により構成されるゼロランを、前記有効桁数符号化部の符号化と異なる周期において符号化し、前記ゼロランを構成する前記組の数の2進表現の桁数より1少ない数の符号0と、前記組の数の2進表現とにより構成される符号を生成するゼロラン符号化部と、

前記ゼロラン以外の係数データについて、前記有効桁数符号化部が符号化した組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数符号化部の符号化と異なる周期において符号化する絶対値符号化部と、

前記ゼロラン以外の係数データについて、前記絶対値符号化部が符号化した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値符号化部の符号化と異なる周期において符号化するサイン符号化部と

を備える画像処理装置。

【請求項2】

前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より1少ない数の2進表現の桁数の符号0と、符号1とにより構成される符号を生成する

10

20

請求項 1 に記載の画像処理装置。

【請求項 3】

前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より 1 少ない数の 2 進表現の桁数の符号 0 により構成される符号を生成する

請求項 1 に記載の画像処理装置。

【請求項 4】

前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期において、前記ゼロランを符号化する

請求項 1 乃至請求項 3 のいずれかに記載の画像処理装置。

10

【請求項 5】

前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期よりも前の周期において、前記ゼロランを符号化する

請求項 1 乃至請求項 4 のいずれかに記載の画像処理装置。

【請求項 6】

画像データをウェーブレット変換するウェーブレット変換部と、
前記ウェーブレット変換部により画像データがウェーブレット変換されて得られたウェーブレット係数を量子化する量子化部と

をさらに備え、

前記有効桁数符号化部、前記ゼロラン符号化部、前記絶対値符号化部、および前記サイン符号化部は、それぞれ、前記量子化部により前記ウェーブレット係数が量子化されて得られた量子化係数について符号化を行う

請求項 1 乃至請求項 5 のいずれかに記載の画像処理装置。

20

【請求項 7】

画像処理装置の画像処理方法であって、
有効桁数符号化部が、画像データから生成された複数の係数データを所定の数毎に 1 組とし、毎周期 1 組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化し、

ゼロラン符号化部が、値が 0 の係数データのみで構成される組により構成されるゼロランを、前記有効桁数の符号化と異なる周期において符号化し、前記ゼロランを構成する前記組の数の 2 進表現の桁数より 1 少ない数の符号 0 と、前記組の数の 2 進表現とにより構成される符号を生成し、

絶対値符号化部が、前記ゼロラン以外の係数データについて、前記有効桁数の符号化が行われた組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数の符号化と異なる周期において符号化し、

サイン符号化部が、前記ゼロラン以外の係数データについて、前記絶対値が符号化された組の前記絶対値が 0 でない各係数データの正負のサインを、前記絶対値の符号化と異なる周期において符号化する

画像処理方法。

【請求項 8】

複数の係数データを所定の数毎に 1 組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期 1 組ずつ、復号する有効桁数復号部と、

前記有効桁数復号部による復号の結果、処理対象である今の組の最大有効桁数が 0 である場合、値が 0 の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数復号部の復号と異なる周期において復号するゼロラン復号部と、

前記ゼロラン以外の係数データについて、前記有効桁数復号部が復号した組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数復号部の復号と異なる周期において復号する絶対値復号部と、

30

40

50

前記ゼロラン以外の係数データについて、前記絶対値復号部が復号した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値復号部の復号と異なる周期において復号するサイン復号部と
を備える画像処理装置。

【請求項9】

前記ゼロラン復号部は、符号1を読み込むか、ラインの最後まで符号を読み込むまで、順次符号を読み込む1回目の読み込みを行い、前記1回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達していない場合、前記1回目の読み込みで読み込んだ符号0の数分の符号をさらに読み込む2回目の読み込みを行い、前記2回目の読み込みで読み込んだ符号の先頭に符号1をつけた符号列を2進表現とする数の、値が0の係数データを生成する

10

請求項8に記載の画像処理装置。

【請求項10】

前記ゼロラン復号部は、前記1回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達した場合、前記ラインの最後まで達する数の、値が0の係数データを生成する

請求項9に記載の画像処理装置。

【請求項11】

前記有効桁数復号部が前記最大有効桁数を示す符号を復号して得られた前記最大有効桁数を、所定の期間保持する最大有効桁数保持部をさらに備え、

20

前記絶対値復号部は、前記最大有効桁数保持部により保持される前記最大有効桁数分の前記絶対値を示す符号を、前記有効桁数復号部による復号が行われた周期から、前記最大有効桁数保持部が前記最大有効桁数を保持した期間分遅延した周期において、復号する

請求項10に記載の画像処理装置。

【請求項12】

前記最大有効桁数保持部は、前記最大有効桁数を、次の周期まで保持する

請求項11に記載の画像処理装置。

【請求項13】

前記最大有効桁数保持部は、前記最大有効桁数を、複数周期後の周期まで保持する

請求項11に記載の画像処理装置。

30

【請求項14】

前記絶対値復号部による復号の結果得られた絶対値、および、前記サイン復号部による復号の結果得られたサインよりなる量子化係数を逆量子化する逆量子化部と、

前記逆量子化部により前記量子化係数が逆量子化されて得られたウェーブレット係数を、ウェーブレット逆変換するウェーブレット逆変換部と

をさらに備える請求項8乃至請求項13のいずれかに記載の画像処理装置。

【請求項15】

画像処理装置の画像処理方法であって、

有効桁数復号部が、複数の係数データを所定の数毎に1組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期1組ずつ、復号し、

40

ゼロラン復号部が、前記最大有効桁数の復号の結果、処理対象である今の組の最大有効桁数が0である場合、値が0の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数の復号と異なる周期において復号し、

絶対値復号部が、前記ゼロラン以外の係数データについて、前記有効桁数が復号された組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数の復号と異なる周期において復号し、

サイン復号部が、前記ゼロラン以外の係数データについて、前記絶対値が復号された組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値の復号と異なる周期において復号する

50

画像処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、画像処理装置および方法に関し、特に、より容易に符号化または復号することができるようにした画像処理装置および方法に関する。

【背景技術】

【0002】

画像（データ）の符号化を行う場合に、入力された画像に対して帯域分割処理を行うことにより生成された各サブバンド（周波数帯域）の係数を符号化する符号化方式として、JPEG（Joint Photographic Experts Group）2000が知られている。

10

【0003】

JPEG2000方式により画像を符号化する場合、入力された画像をウェーブレット変換することにより得られたウェーブレット係数が量子化され、量子化により得られた量子化係数がさらにエントロピ符号化される。

【0004】

従来、エントロピ符号化においては、EBCOT（Embedded Block Coding with Optimized Truncation）と称されるビットモデリングと、MQコードと称される算術符号化が行われている。すなわち、量子化係数がビットモデリングされ、さらにビットプレーンごとに複数のコーディングパスに基づいて算術符号化される。そして、算術符号化により得られた符号が符号化された画像（データ）として出力される（例えば、特許文献1参照）。

20

【0005】

また、JPEG2000方式により符号化された画像を復号する場合、符号化を行う場合とは逆の手順で処理が行われる。より具体的には、符号化された画像のデータとしての符号がエントロピ復号されて逆量子化され、これにより得られた量子化係数がさらにウェーブレット逆変換される。そして、ウェーブレット逆変換により得られた画像が復号された画像として出力される。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2004-166254号公報

30

【発明の概要】

【発明が解決しようとする課題】

【0007】

しかしながら上述した技術では、EBCOTおよびMQコードにおける処理量が多いため、画像の符号化および復号を高速に行うことは困難であり、例えば、横1920画素、縦1080画素などの解像度の高いHD（High Definition）画像をリアルタイムで符号化（または復号）するためには、高価な専用のハードウェアを用意する必要があった。

【0008】

本開示は、このような状況に鑑みてなされたものであり、画像をより容易に符号化または復号することができるようにするものである。

40

【課題を解決するための手段】

【0009】

本開示の一側面は、画像データから生成された複数の係数データを所定の数毎に1組とし、毎周期1組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化する有効桁数符号化部と、値が0の係数データのみで構成される組により構成されるゼロランを、前記有効桁数符号化部の符号化と異なる周期において符号化し、前記ゼロランを構成する前記組の数の2進表現の桁数より1少ない数の符号0と、前記組の数の2進表現とにより構成される符号を生成するゼロラン符号化部と、前記ゼロラン以外の係数データについて、前記有効桁数符号化部が符号化し

50

た組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数符号化部の符号化と異なる周期において符号化する絶対値符号化部と、前記ゼロラン以外の係数データについて、前記絶対値符号化部が符号化した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値符号化部の符号化と異なる周期において符号化するサイン符号化部とを備える画像処理装置である。

【0011】

前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より1少ない数の2進表現の桁数の符号0と、符号1とにより構成される符号を生成することができる。

【0012】

前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より1少ない数の2進表現の桁数の符号0により構成される符号を生成することができる。

【0013】

前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期において、前記ゼロランを符号化することができる。

前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期よりも前の周期において、前記ゼロランを符号化することができる。

【0014】

画像データをウェーブレット変換するウェーブレット変換部と、前記ウェーブレット変換部により画像データがウェーブレット変換されて得られたウェーブレット係数を量子化する量子化部とをさらに備え、前記有効桁数符号化部、前記ゼロラン符号化部、前記絶対値符号化部、および前記サイン符号化部は、それぞれ、前記量子化部により前記ウェーブレット係数が量子化されて得られた量子化係数について符号化を行うことができる。

【0015】

本開示の一側面は、また、画像処理装置の画像処理方法であって、有効桁数符号化部が、画像データから生成された複数の係数データを所定の数毎に1組とし、毎周期1組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化し、ゼロラン符号化部が、値が0の係数データのみで構成される組により構成されるゼロランを、前記有効桁数の符号化と異なる周期において符号化し、前記ゼロランを構成する前記組の数の2進表現の桁数より1少ない数の符号0と、前記組の数の2進表現とにより構成される符号を生成し、絶対値符号化部が、前記ゼロラン以外の係数データについて、前記有効桁数の符号化が行われた組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数の符号化と異なる周期において符号化し、サイン符号化部が、前記ゼロラン以外の係数データについて、前記絶対値が符号化された組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値の符号化と異なる周期において符号化する画像処理方法である。

【0016】

本開示の他の側面は、複数の係数データを所定の数毎に1組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期1組ずつ、復号する有効桁数復号部と、前記有効桁数復号部による復号の結果、処理対象である今の組の最大有効桁数が0である場合、値が0の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数復号部の復号と異なる周期において復号するゼロラン復号部と、前記ゼロラン以外の係数データについて、前記有効桁数復号部が復号した組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数復号部の復号と異なる周期において復号する絶対値復号部と、前記ゼロラン以外の係数データについて、前記絶対値復号部が復号した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値復号部の復号と異なる周期において復号するサイン復号部とを備え

10

20

30

40

50

る画像処理装置である。

【 0 0 1 8 】

前記ゼロラン復号部は、符号 1 を読み込むか、ラインの最後まで符号を読み込むまで、順次符号を読み込む 1 回目の読み込みを行い、前記 1 回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達していない場合、前記 1 回目の読み込みで読み込んだ符号 0 の数分の符号をさらに読み込む 2 回目の読み込みを行い、前記 2 回目の読み込みで読み込んだ符号の先頭に符号 1 をつけた符号列を 2 進表現とする数の、値が 0 の係数データを生成することができる。

【 0 0 1 9 】

前記ゼロラン復号部は、前記 1 回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達した場合、前記ラインの最後まで達する数の、値が 0 の係数データを生成することができる。

【 0 0 2 0 】

前記有効桁数復号部が前記最大有効桁数を示す符号を復号して得られた前記最大有効桁数を、所定の期間保持する最大有効桁数保持部をさらに備え、前記絶対値復号部は、前記最大有効桁数保持部により保持される前記最大有効桁数分の前記絶対値を示す符号を、前記有効桁数復号部による復号が行われた周期から、前記最大有効桁数保持部が前記最大有効桁数を保持した期間分遅延した周期において、復号することができる。

【 0 0 2 1 】

前記最大有効桁数保持部は、前記最大有効桁数を、次の周期まで保持することができる。

【 0 0 2 2 】

前記最大有効桁数保持部は、前記最大有効桁数を、複数周期後の周期まで保持することができる。

【 0 0 2 3 】

前記絶対値復号部による復号の結果得られた絶対値、および、前記サイン復号部による復号の結果得られたサインよりなる量子化係数を逆量子化する逆量子化部と、前記逆量子化部により前記量子化係数が逆量子化されて得られたウェーブレット係数を、ウェーブレット逆変換するウェーブレット逆変換部とをさらに備えることができる。

【 0 0 2 4 】

本開示の他の側面は、また、画像処理装置の画像処理方法であって、有効桁数復号部が、複数の係数データを所定の数毎に 1 組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期 1 組ずつ、復号し、ゼロラン復号部が、前記最大有効桁数の復号の結果、処理対象である今の組の最大有効桁数が 0 である場合、値が 0 の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数の復号と異なる周期において復号し、絶対値復号部が、前記ゼロラン以外の係数データについて、前記有効桁数が復号された組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数の復号と異なる周期において復号し、サイン復号部が、前記ゼロラン以外の係数データについて、前記絶対値が復号された組の前記絶対値が 0 でない各係数データの正負のサインを、前記絶対値の復号と異なる周期において復号する画像処理方法である。

【 0 0 2 5 】

本開示の一側面においては、画像データから生成された複数の係数データを所定の数毎に 1 組とし、毎周期 1 組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数が求められ、最大有効桁数に関する情報が符号化され、値が 0 の係数データのみで構成される組により構成されるゼロランが、有効桁数の符号化と異なる周期において符号化され、そのゼロランを構成する組の数の 2 進表現の桁数より 1 少ない数の符号 0 と、その組の数の 2 進表現とにより構成される符号が生成され、ゼロラン以外の係数データについて、有効桁数の符号化が行われた組の各係数データの最大有効桁数分の絶対値が抽出され、絶対値が、有効桁数の符号化と異なる周期において符号化され、ゼロラン以外の係数デ

10

20

30

40

50

ータについて、絶対値が符号化された組の絶対値が0でない各係数データの正負のサインが、絶対値の符号化と異なる周期において符号化される。

【0026】

本開示の他の側面においては、複数の係数データを所定の数毎に1組とし、組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号が、毎周期1組ずつ、復号され、その最大有効桁数の復号の結果、処理対象である今の組の最大有効桁数が0である場合、値が0の係数データのみで構成される組により構成されるゼロランを示す符号が、有効桁数の復号と異なる周期において復号され、ゼロラン以外の係数データについて、有効桁数が復号された組の各係数データの最大有効桁数分の絶対値を示す符号が、有効桁数の復号と異なる周期において復号され、ゼロラン以外の係数データについて、絶対値が復号された組の絶対値が0でない各係数データの正負のサインが、絶対値の復号と異なる周期において復号される。

10

【発明の効果】

【0027】

本開示によれば、画像を処理することが出来る。特に、画像を、より容易に符号化または復号することができる。

【図面の簡単な説明】

【0028】

【図1】画像符号化装置の主な構成例を示すブロック図である。

【図2】サブバンドを説明するための図である。

20

【図3】符号化される量子化係数の一例を示す図である。

【図4】ゼロラン符号化の一例を示す図である。

【図5】ゼロランの符号の一例を説明する図である。

【図6】ゼロラン符号化の、他の例を示す図である。

【図7】ゼロランの符号の、他の例を説明する図である。

【図8】ゼロラン符号化の、さらに他の例を示す図である。

【図9】ゼロランの符号の、さらに他の例を説明する図である。

【図10】パイプライン処理を行わない場合のタイミングチャートの例を説明する図である。

【図11】非パイプライン処理の符号順序の例を説明する図である。

30

【図12】パイプライン処理を行う場合のタイミングチャートの例を説明する図である。

【図13】パイプライン処理を行う場合のタイミングチャートの、他の例を説明する図である。

【図14】パイプライン処理の符号順序の例を説明する図である。

【図15】パイプライン処理の符号順序の、他の例を説明する図である。

【図16】エントロピ符号化部の主な構成例を示すブロック図である。

【図17】VLC符号化部の主な構成例を示すブロック図である。

【図18】符号化処理の流れの例を説明するフローチャートである。

【図19】エントロピ符号化処理の流れの例を説明するフローチャートである。

【図20】W個組符号化処理の流れの例を説明するフローチャートである。

40

【図21】最大有効桁数処理の流れの例を説明するフローチャートである。

【図22】最大有効桁数処理の流れの例を説明する、図21に続くフローチャートである。

【図23】画像復号装置の主な構成例を示すブロック図である。

【図24】エントロピ復号部の主な構成例を示すブロック図である。

【図25】符号分割部の構成例を示すブロック図である。

【図26】符号分割部の、他の構成例を示すブロック図である。

【図27】VLC復号部の主な構成例を示すブロック図である。

【図28】復号処理の流れの例を説明するフローチャートである。

【図29】エントロピ復号処理の流れの例を説明するフローチャートである。

50

【図 3 0】W個組復号処理の流れの例を説明するフローチャートである。

【図 3 1】最大有効桁数処理の流れの例を説明するフローチャートである。

【図 3 2】最大有効桁数処理の流れの例を説明する、図 3 1 に続くフローチャートである。

【図 3 3】最大有効桁数処理の流れの他の例を説明する、図 2 1 に続くフローチャートである。

【図 3 4】最大有効桁数処理の流れの、他の例を説明するフローチャートである。

【図 3 5】最大有効桁数処理の流れの、他の例を説明する、図 3 4 に続くフローチャートである。

【図 3 6】パイプライン処理を行う場合のタイミングチャートの、さらに他の例を説明する図である。

10

【図 3 7】エントロピ符号化部の他の構成例を示すブロック図である。

【図 3 8】エントロピ復号部の他の構成例を示すブロック図である。

【図 3 9】画像符号化装置の他の構成例を示すブロック図である。

【図 4 0】画像復号装置の他の構成例を示すブロック図である。

【図 4 1】符号化・復号の流れの例を説明する図である。

【図 4 2】パーソナルコンピュータの主な構成例を示すブロック図である。

【発明を実施するための形態】

【0029】

以下、本開示を実施するための形態（以下実施の形態とする）について説明する。なお、説明は以下の順序で行う。

20

1. 第 1 の実施の形態（画像符号化装置）
2. 第 2 の実施の形態（画像復号装置）
3. 第 3 の実施の形態（画像符号化装置・画像復号装置）
4. 第 4 の実施の形態（画像符号化装置・画像復号装置）
5. 第 5 の実施の形態（パーソナルコンピュータ）

【0030】

< 1. 第 1 の実施の形態 >

[画像符号化装置]

図 1 は、画像符号化装置の主な構成例を示すブロック図である。図 1 に示される画像符号化装置 1 1 は、画像データを符号化し、得られた符号化データを出力する画像処理装置である。

30

【0031】

図 1 に示されるように、画像符号化装置 1 1 は、ウェーブレット変換部 2 1、量子化部 2 2、およびエントロピ符号化部 2 3 を有する。

【0032】

ウェーブレット変換部 2 1 には、例えば、必要に応じて DC レベルシフトが施されたコンポーネント信号である画像（データ）が入力される。ウェーブレット変換部 2 1 は、その入力された画像をウェーブレット変換して複数のサブバンドに分解する。ウェーブレット変換部 2 1 は、ウェーブレット変換により得られたサブバンドのウェーブレット係数を量子化部 2 2 に供給する。

40

【0033】

量子化部 2 2 は、ウェーブレット変換部 2 1 から供給されたウェーブレット係数を量子化し、その結果として得られた量子化係数をエントロピ符号化部 2 3 に供給する。

【0034】

エントロピ符号化部 2 3 は、量子化部 2 2 から供給された量子化係数をエントロピ符号化し、これにより得られた符号を符号化された画像（符号化データ）として出力する。エントロピ符号化部 2 3 から出力された画像（符号化データ）は、例えば、レート制御処理された後、パケット化されて記録されたり、画像符号化装置 1 1 に接続された他の装置（図示せず）に供給されたりする。

50

【 0 0 3 5 】

エントロピ符号化部 2 3 は、以下に説明するように量子化係数の符号化を行う。これにより、エントロピ符号化部 2 3 は、より容易な符号化を実現する。

【 0 0 3 6 】

また、ウェーブレット変換部 2 1 乃至エントロピ符号化部 2 3 の各処理部は、以下に説明するようなタイミングで、それぞれの処理を行う。これにより、画像符号化装置 1 1 は、より高速な符号化を実現する。

【 0 0 3 7 】

さらに、各処理部は、以下に説明するように、値が 0 の量子化係数よりなるゼロランをまとめて符号化する。これにより、画像符号化装置 1 1 は、符号化効率を向上させることができる。

【 0 0 3 8 】

次に、図 2 および図 3 を参照して、図 1 のエントロピ符号化部 2 3 が行うエントロピ符号化について説明する。

【 0 0 3 9 】

図 2 は、サブバンドを説明するための図である。例えば、図 2 に示すように、1 つのサブバンドがライン L1 乃至ライン L6 の 6 本のラインから構成されているとし、xy 座標系におけるライン上の画素に対応する位置を (x, y) とする。ここで、各ラインの図中、左端の位置の x 座標は 0 とされ、ライン L1 の y 座標は 0 とされる。

【 0 0 4 0 】

量子化部 2 2 からエントロピ符号化部 2 3 には、ビットプレーン表現された、サブバンドの各位置 (x, y) における量子化係数がライン L1 からライン L6 までラスタスキャン順に入力される。

【 0 0 4 1 】

換言するに、エントロピ符号化部 2 3 には、まず、ライン L1 の左端の位置 $(0, 0)$ に対応する量子化係数が入力される。次にその位置 $(0, 0)$ の右隣の位置 $(1, 0)$ に対応する量子化係数がエントロピ符号化部 2 3 に入力され、ライン L1 の右端の位置まで、量子化係数が入力された位置の右隣の位置に対応する量子化係数がエントロピ符号化部 2 3 に順次入力される。

【 0 0 4 2 】

そして、ライン L1 上の位置の量子化係数が全て入力されると、ライン L2 の左端の位置 $(0, 1)$ から順番に右端の位置まで、ライン L2 上の各位置に対応する量子化係数がエントロピ符号化部 2 3 に入力され、同様にライン L3 からライン L6 まで、各ライン上の位置に対応する量子化係数がエントロピ符号化部 2 3 に入力される。

【 0 0 4 3 】

例えば、図 3 の図中、左上に示すように、図 2 のライン L1 の左端の位置の量子化係数から順番に、12 個の量子化係数がエントロピ符号化部 2 3 に入力されると、エントロピ符号化部 2 3 は、予め定められた所定の数 w (図 3 では $w = 4$) ずつ量子化係数を符号化する。

【 0 0 4 4 】

ここで、図 3 の左上に示された各量子化係数は、その符号の絶対値が 2 進数の桁に分けられて表現されて (ビットプレーン表現されて) おり、図 3 の例では、エントロピ符号化部 2 3 には、1 つのライン (図 2 のライン L1) の量子化係数 “-0101”、“+0011”、“-0110”、“+0010”、“+0011”、“+0110”、“0000”、“-0011”、“+1101”、“-0100”、“+0111”、および “-1010” が順番に入力される。

【 0 0 4 5 】

1 つの量子化係数は、“+” (正) または “-” (負) で表わされる量子化係数の符号 (以下、量子化係数のサイン (Sign) と称する。) と、2 進数で表わされた量子化係数の絶対値とからなる。図 3 では、量子化係数の絶対値の各桁の値を示す各ビットのうち、図中、最も上側のビットが最上位ビット (最上位の桁のビット) を表わしている。したがって、例えば、量子化係数 “-0101” は、そのサインが “-” であり、2 進数で表わされた

10

20

30

40

50

絶対値が“0101”であるので、この量子化係数は10進数で表わすと“-5”となる。

【0046】

まず、エントロピ符号化部23は、入力された1つのラインの量子化係数(の絶対値)が全て0(ゼロ)であるか否かを判定し、その判定結果に応じて、これから符号化するラインの量子化係数が全て0であるか否かを示す符号を出力する。量子化係数が全て0であると判定した場合、エントロピ符号化部23は、ラインの量子化係数が全て0であるか否かを示す符号として0を出力して、現在行っているラインの量子化係数の符号化を終了する。また、全ての量子化係数の値が0ではない(0の量子化係数のみではない)と判定した場合、エントロピ符号化部23は、ラインの量子化係数が全て0であるか否かを示す符号として1を出力する。

10

【0047】

図中、左上に示した12個の量子化係数が入力された場合、入力されたラインの量子化係数は0のみではないので、図中、右上に示すように、エントロピ符号化部23は符号として1を出力する。

【0048】

ラインの量子化係数が全て0であるか否かを示す符号として、量子化係数が全て0でないことを示す符号1が出力されると、次に、エントロピ符号化部23は、入力された最初の4つ(W個)の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の符号化を行う。

【0049】

エントロピ符号化部23は、今回入力された連続する4つの量子化係数の最大有効桁数(図3における変数Bの値)と、前回符号化した(入力された)4つ(W個)の量子化係数の最大有効桁数とを比較し、最大有効桁数が変化したか否かを判定して、量子化係数の最大有効桁数を示す符号を出力する。

20

【0050】

ここで、最大有効桁数とは、まとめて符号化する4つ(W個)の量子化係数のうち、絶対値が最も大きい量子化係数の有効桁数をいう。換言すれば、最大有効桁数は、4つの量子化係数のうち、絶対値が最も大きい量子化係数の最上位にある1が何桁目にあるかを示す。したがって、例えば、まとめて符号化する4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数は、絶対値が最も大きい量子化係数“-0110”の最上位にある1の桁である“3”とされる。

30

【0051】

また、量子化係数の最大有効桁数を示す符号は、最大有効桁数が変化したか否かを示す符号、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号からなり、最大有効桁数が変化していない場合、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号は出力されない。

【0052】

エントロピ符号化部23は、最大有効桁数の比較の結果、最大有効桁数が変化した場合、最大有効桁数が変化したことを示す符号1を出力し、最大有効桁数が変化していない場合、最大有効桁数が変化していないことを示す符号0を出力する。

40

【0053】

なお、最大有効桁数が変化したか否かを判定する場合に、今回初めて4つの量子化係数が入力される時、すなわち、符号化するサブバンドの量子化係数が初めて入力される時(例えば、図2のラインL1の左端から順番に4つの量子化係数が入力される時)、前回、そのサブバンドの量子化係数は符号化されていないので、前回符号化した4つ(W個)の量子化係数の最大有効桁数は0とされる。

【0054】

したがって、エントロピ符号化部23は、今回入力された4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数3と、前回符号化した量子化

50

係数の最大有効桁数 0 とを比較し、最大有効桁数が変化したので符号 1 を出力する。

【 0 0 5 5 】

また、エントロピ符号化部 2 3 は、最大有効桁数が変化したことを示す符号 1 に続いて、最大有効桁数が増加したか、または減少したかを示す符号を出力する。ここで、エントロピ符号化部 2 3 は、最大有効桁数が増加した場合には 0 を出力し、最大有効桁数が減少した場合には 1 を出力する。

【 0 0 5 6 】

図 3 の例の場合、前回の最大有効桁数は 0 であり、今回の最大有効桁数は 3 であるので、図中、右上に示すように、エントロピ符号化部 2 3 は、最大有効桁数が増加したことを示す符号 0 を出力する。

10

【 0 0 5 7 】

さらに、エントロピ符号化部 2 3 は、最大有効桁数が増加したか、または減少したかを示す符号を出力すると、最大有効桁数がどれだけ増加または減少したかを示す符号、すなわち、最大有効桁数の変化量を示す符号を出力する。具体的には、エントロピ符号化部 2 3 は、最大有効桁数の変化量（すなわち、増加量または減少量）を n とすると、 $(n-1)$ 個の符号 0 を出力し、それらの 0 に続いて符号 1 を出力する。

【 0 0 5 8 】

図 3 の最初の 4 つの量子化係数を符号化する場合、最大有効桁数の変化量は 3 ($=3-0$) であるので、エントロピ符号化部 2 3 は符号として、2 ($=3-1$) 個の 0 を出力し、さらに 1 を出力する。

20

【 0 0 5 9 】

次に、エントロピ符号化部 2 3 は、今回符号化する 4 つ (W 個) の量子化係数のそれぞれの絶対値を示す最大有効桁数分の符号を出力する。すなわち、エントロピ符号化部 2 3 は、それぞれの量子化係数について、最大有効桁数により示される有効桁の最大の桁から順番に最小の桁まで、量子化係数の絶対値の各桁の値を示す符号を出力する。

【 0 0 6 0 】

今回符号化する量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”であるので、エントロピ符号化部 2 3 は、まず、最初に入力された量子化係数“-0101”の絶対値を示す最大有効桁数分の符号を出力する。ここで、今回の最大有効桁数は 3 であるので、エントロピ符号化部 2 3 は、量子化係数“-0101”の最大有効桁数により示される有効桁の最大の桁（すなわち、3 桁目）の値“1”、最大の桁より 1 つ下の桁（2 桁目）の値“0”、および最下位の桁の値“1”を出力する。これにより、量子化係数“-0101”の絶対値を示す有効桁数分の符号“101”が出力される。

30

【 0 0 6 1 】

同様に、エントロピ符号化部 2 3 は、量子化係数“+0011”、“-0110”、および“+0010”の絶対値を示す有効桁数分の符号“011”、“110”、および“010”を順番に出力する。したがって、量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”のそれぞれの絶対値を示す最大有効桁数分の符号として、“101011110010”が出力される。このように、エントロピ符号化部 2 3 からは、量子化係数の絶対値を示す符号として、符号化する 4 つの量子化係数の最大有効桁数に応じた長さの符号が出力される。

40

【 0 0 6 2 】

そして最後に、エントロピ符号化部 2 3 は、4 つ (W 個) の量子化係数のうち、絶対値が 0 でない量子化係数のそれぞれのサインを示す符号を出力する。ここで、エントロピ符号化部 2 3 は、量子化係数のサインが“+”（正）である場合、符号 0 を出力し、サインが“-”（負）である場合、符号 1 を出力する。

【 0 0 6 3 】

今回符号化する量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”であり、これらの量子化係数のサインは、順に負、正、負、正であるので、図中、右上に示すように、エントロピ符号化部 2 3 は、量子化係数のそれぞれのサインを示す符号として、“1010”を出力する。

50

【 0 0 6 4 】

最初に入力された4つの量子化係数が符号化されると、エントロピ符号化部23は、続いて、次の連続する4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”の符号化を行う。

【 0 0 6 5 】

最初に(前回)入力された量子化係数の符号化における場合と同様に、まず、エントロピ符号化部23は、今回、新たに入力された4つ(W個)の量子化係数の最大有効桁数と、前回符号化した4つの量子化係数の最大有効桁数とを比較する。

【 0 0 6 6 】

今回入力された4つ(W個)の量子化係数“+0011”、“+0110”、“0000”、および“-0011”の最大有効桁数は、絶対値が最も大きい量子化係数“+0110”の最上位にある1の桁である“3”であり、前回符号化した量子化係数の最大有効桁数“3”と同じであるので、エントロピ符号化部23は、最大有効桁数が変化していないことを示す符号0を出力する。

10

【 0 0 6 7 】

続いて、エントロピ符号化部23は、今回符号化する4つ(W個)の量子化係数“+0011”、“+0110”、“0000”、および“-0011”のそれぞれの絶対値を示す最大有効桁数分の符号“011”、“110”、“000”、および“011”が順番に並べられた符号“011110000011”を出力する。

【 0 0 6 8 】

そして、量子化係数の絶対値を示す符号が出力されると、エントロピ符号化部23は、4つの量子化係数のうち、絶対値が0でない量子化係数のそれぞれのサインを示す符号を出力する。

20

【 0 0 6 9 】

今回符号化する量子化係数は、“+0011”、“+0110”、“0000”、および“-0011”であり、3つ目の量子化係数“0000”はその絶対値が0であるので、エントロピ符号化部23は、0でない量子化係数“+0011”、“+0110”、および“-0011”のそれぞれのサイン(正、正、負)を示す符号“001”を出力する。

【 0 0 7 0 】

4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”が符号化されると、さらに、エントロピ符号化部23は、次の4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の符号化を行う。

30

【 0 0 7 1 】

まず、エントロピ符号化部23は、今回、新たに入力された4つ(W個)の量子化係数の最大有効桁数と、前回符号化した4つの量子化係数の最大有効桁数とを比較する。

【 0 0 7 2 】

今回入力された4つ(W個)の量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の最大有効桁数は、絶対値が最も大きい量子化係数“+1101”の最上位にある1の桁である“4”であり、前回符号化した量子化係数の最大有効桁数“3”とは異なるので、エントロピ符号化部23は、最大有効桁数が変化したことを示す符号1を出力する。

40

【 0 0 7 3 】

また、前回の最大有効桁数は3であり、今回の最大有効桁数は4であるので、エントロピ符号化部23は、図中、右側に示すように、最大有効桁数が増加したことを示す符号0を出力する。

【 0 0 7 4 】

さらに、エントロピ符号化部23は、最大有効桁数がどれだけ増加または減少したかを示す符号を出力する。この場合、最大有効桁数の変化量は1(=4-3)であるので、エントロピ符号化部23は、符号として、0(=1-1)個の0を出力し、さらに1を出力する(すなわち、符号1を出力する)。

【 0 0 7 5 】

50

次に、エントロピ符号化部 2 3 は、今回符号化する 4 つ (W 個) の量子化係数 “ +1101 ”、“ -0100 ”、“ +0111 ”、および “ -1010 ” のそれぞれの絶対値を示す最大有効桁数分の符号 “ 1101 ”、“ 0100 ”、“ 0111 ”、および “ 1010 ” が順番に並べられた符号 “ 1101010001111010 ” を出力する。

【 0 0 7 6 】

そして、量子化係数の絶対値を示す符号が出力されると、エントロピ符号化部 2 3 は、4 つの量子化係数のうち、0 でない量子化係数のそれぞれのサインを示す符号を出力する。

【 0 0 7 7 】

今回符号化する量子化係数は、“ +1101 ”、“ -0100 ”、“ +0111 ”、および “ -1010 ” であり、これらの量子化係数のサインは、順番に正、負、正、負であるので、図中、右下に示すように、エントロピ符号化部 2 3 は、量子化係数のそれぞれのサインを示す符号として、“ 0101 ” を出力する。

10

【 0 0 7 8 】

このようにして、エントロピ符号化部 2 3 は、入力された量子化係数を、連続する予め定められた数 (W 個) ずつ符号化する。これにより、エントロピ符号化部 2 3 からは、符号化するラインの量子化係数が全て 0 であるか否かを示す符号が出力され、ラインの量子化係数が全て 0 でないことを示す符号が出力されると、次に、W 個の量子化係数の最大有効桁数を示す符号、W 個の量子化係数の絶対値 (ビットプレーン表現) を示す符号、およびそれらの量子化係数のサインを示す符号が出力される。

20

【 0 0 7 9 】

そして、これらの W 個の量子化係数の最大有効桁数を示す符号、W 個の量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号のそれぞれは、そのラインの量子化係数が全て符号化されるまで、次の W 個の量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号が繰り返し出力される。

【 0 0 8 0 】

なお、量子化係数がラスタスキャン順に符号化されると説明したが、量子化係数が符号化される順番は必ずしもラスタスキャン順である必要はない。例えば、図 2 に示したサブバンドの量子化係数が符号化される場合、最初に位置 (0 , 0)、(0 , 1)、(0 , 2)、および (0 , 3) (すなわち、ライン L1 乃至ライン L4 のそれぞれの図中、左端の位置) の量子化係数が符号化され、次に、位置 (1 , 0)、(1 , 1)、(1 , 2)、および (1 , 3) の量子化係数が符号化されるといったように、図中、縦方向に並ぶ 4 つの位置の量子化係数を W 個の量子化係数として、W 個ずつ順番に符号化するようにしてもよい。

30

【 0 0 8 1 】

このようにして、エントロピ符号化部 2 3 は、サブバンドの量子化係数を、予め定められた所定の数ずつまとめて符号化し、量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号を出力する。

【 0 0 8 2 】

このように、サブバンドの量子化係数を、予め定められた所定の数ずつまとめて符号化することで、例えば、JPEG2000 方式により画像を符号化する場合とは異なり、複数のコーディングパスに基づいて、ビットプレーンごとに複数の処理を行う必要がなく、また、可変長符号化を行うので、符号化の処理量を大幅に低減することができる。これにより、より高速に画像の符号化を行うことができ、高解像度の画像をリアルタイムで符号化するための符号化装置を安価で実現することができる。

40

【 0 0 8 3 】

さらに、画像符号化装置 1 1 においては、画像を符号化する場合に、符号の長さを明示的に符号化する必要がないため、符号量を少なくすることができ、符号の長さの情報を管理する必要もない。

【 0 0 8 4 】

50

なお、以上においては、W個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数を、最大有効桁数を示す変数Bnewの値とすると説明したが、変数Bnewの値は、W個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数以上の値であればよい。変数Bnewの値が大きくなると、量子化係数の絶対値を示す符号の符号量が多くなるが、変数Bnewの値を、絶対値の最も大きい量子化係数の有効桁数以上の値とすることで、量子化係数の最大有効桁数を示す符号の符号量を少なくすることができる。

【0085】

[ゼロラン処理1]

以上のような符号化方法の場合、処理対象のラインの量子化係数が全て0でなければ、値が0の量子化係数も、値が0でない量子化係数の場合と同様に処理される。しかしながら、値が0の量子化係数は、局所的に集中し易く、連続して出現することが多い。

10

【0086】

そこで、エントロピ符号化部23は、このような値が0の量子化係数が連続して出現する部分(0連長若しくはゼロランとも称する)を、まとめて符号化する。このようにすることにより、エントロピ符号化部23は、このような0連長(ゼロラン)の部分で、値が0でない量子化係数の組を符号化する場合よりも効率よく符号化することができる。つまり、エントロピ符号化部23は、0が多い係数データの符号化の符号化効率を向上させることができる。

【0087】

以下に、より具体的に説明する。図4は、ゼロラン符号化の一例を示す図である。

20

【0088】

例えば、図4の上を示されるように、処理対象のラインにおいて、4つ(W個)の量子化係数の組がM組連続するとする。この処理対象のラインには、値が0でない量子化係数も含まれるので、最初の符号は1となる。

【0089】

図4中、最初に処理される一番左の、4つ(W個)の量子化係数の組の最大有効桁数は3であるので、次に、最大有効桁数Bが初期値0から「変更あり(diff)」であることを示す符号1が出力される。次に、「Bを増やす(plus)」ことを示す符号0が出力される。

【0090】

最大有効桁数Bの値は0から3に増える(増分3)ので、次に、符号0が2つ(増分3-1)出力され、続いて、変動の終端(enough)を示す符号1が出力される。つまり、この符号1が検出されることにより、変更後の最大有効桁数Bの値が決定される。

30

【0091】

そして、今回符号化する4つ(W個)の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”のそれぞれの絶対値を示す最大有効桁数分の符号“101011110010”が出力される。

【0092】

さらに、今回符号化する4つ(W個)の量子化係数の内の、0でない量子化係数の符号(+か-か)を示す符号“1010”が出力される。

40

【0093】

そして、符号化処理対象は、次の4つ(W個)の量子化係数の組に移る。図4に示されるように、左から2番目の、4つ(W個)の量子化係数の組は、その量子化係数の値が全て0である。つまり、最大有効桁数Bの値は、0である。そして、この最大有効桁数B=0の組が、M組連続する(0連長M)(Mは任意の自然数)。

【0094】

エントロピ符号化部23は、このようなゼロラン(0連長)を、以下のようにまとめて符号化する。

【0095】

まず、最大有効桁数B=0の組の場合と同様に、最大有効桁数Bが0以外の値(例えば

50

3) から 0 に変わるので、「変更あり (diff)」であることを示す符号 1 が出力される。次に、「B を減らす (minus)」ことを示す符号 1 が出力される。

【0096】

次に、最大有効桁数 B の値は 3 から 0 に減る (減分 3) ので、次に、符号 0 が 2 つ (減分 3 - 1) 出力される。

【0097】

ここで、前の組と同様、変動の終端 (enough) を示す符号 1 が出力されるようにしてもよいが、この組は最大有効桁数 B の変更後の値が 0 であるため、変動の終端 (enough) を示す符号 1 の出力が省略されるようにしてもよい。

【0098】

最大有効桁数 B の減分を示す符号 (図 4 の例では符号 0 が 2 つ) によって、最大有効桁数 B の値が 0 になることが明らかとなる。最大有効桁数 B の最小値は、0 であり、これ以上は減少しない。つまり、この 2 つの符号 0 によって、最大有効桁数 B の変更後の値が 0 に確定する。

【0099】

したがって、このように最大有効桁数 B の変更後の値が 0 である場合、変動の終端 (enough) を示す符号 1 は不要であり、省略することができる。このようにすることにより、エントロピ符号化部 23 は、符号 1 の出力を省略した分、符号量を低減させることができる。すなわち、エントロピ符号化部 23 は、符号化効率を向上させることができる。

【0100】

次に、ゼロランを示す符号が出力される。ゼロランの長さ (ゼロ連長) M (組) を 2 進表現した場合の桁数 N から 1 を減算した値の数分の符号 0 の後に 1 つの符号 1 を並べ、さらに、N 桁の 2 進数を確定するための、0 連長の 2 進表現の最上位ビット以外の各桁の値を並べたものが、このゼロランを示す符号とされる。

【0101】

なお、この符号に、最上位ビットを含めないのは、その値が必ず 1 だからである。換言するに、(N - 1) 個の符号 0 の後に、0 連長の 2 進表現を並べたものが、このゼロランを示す符号とされているとも言える。つまり、この符号化は、0 連長の 2 進表現を、0 連長の桁数 - 1 だけシフトすることで、容易に実現することができる。

【0102】

例えば、エントロピ符号化部 23 は、各 0 連長に対して、図 5 に示されるような表 (テーブル情報) を予め記憶しておく。エントロピ符号化部 23 は、ゼロランの長さ (0 連長) M を求め、図 5 のテーブル情報を用いて、その M に対応する符号を選択し、その符号を出力する。

【0103】

なお、図 5 においては、説明の便宜上、符号にカンマ (,) が設けられているが、実際の符号には、このカンマは、含まれない。

【0104】

以上のように、エントロピ符号化部 23 は、ゼロランをまとめて、より効率よく符号化することができる。つまり、エントロピ符号化部 23 は、0 が多い係数データの符号化の符号化効率を向上させることができる。

【0105】

以上のようにゼロランが符号化されると、処理対象が、図 4 の一番右の、4 つ (W 個) の量子化係数の組に移される。

【0106】

この組の最大有効桁数 B は 4 である。したがって、最大有効桁数 B を示す符号が出力される。ただし、ここで、最大有効桁数 B の値は 0 から 4 に変わるので、最大有効桁数 B を示す符号として、上述したように、「変更あり (diff)」であることを示す符号 1 や、「B を増やす (plus)」ことを示す符号 0 が出力されるようにしてもよいが、前の組の最大有効桁数 B が 0 であることから、これらの符号の出力が省略されるようにしてもよい。

10

20

30

40

50

【 0 1 0 7 】

最大有効桁数 B の最小値は 0 であるので、前の組の最大有効桁数 B が 0 である場合、処理対象であるこの組の最大有効桁数 B が変化するのであれば、それは増加方向であることが明らかである。また、この組が最大有効桁数 B が変化しない (0 のままである) ののであれば、この組もゼロランに含まれるので、最大有効桁数 B を示す符号は出力されない。

【 0 1 0 8 】

したがって、この組の最大有効桁数 B が 0 でなく、かつ、前の組の最大有効桁数 B が 0 である場合、「変更あり (diff) 」であることを示す符号 1 や、「 B を増やす (plus) 」ことを示す符号 0 の出力が省略されるようにすることができる。このようにすることにより、エントロピ符号化部 2 3 は、符号の出力を省略した分、符号量を低減させることができる。すなわち、エントロピ符号化部 2 3 は、符号化効率を向上させることができる。

10

【 0 1 0 9 】

次に、最大有効桁数 B の値が 0 から 4 に増える (増分 4) ので、次に、符号 0 が 3 つ (増分 4 - 1) 出力され、続いて、変動の終端 (enough) を示す符号 1 が出力される。つまり、この符号 1 が検出されることにより、変更後の最大有効桁数 B の値が決定される。

【 0 1 1 0 】

そして、今回符号化する 4 つ (W 個) の量子化係数 “ +1101 ”、“ -0100 ”、“ +0111 ”、および “ -1010 ” のそれぞれの絶対値を示す最大有効桁数分の符号 “ 1101010001111010 ” が出力される。

【 0 1 1 1 】

さらに、今回符号化する 4 つ (W 個) の量子化係数の内の、0 でない量子化係数の符号 (+ か - か) を示す符号 “ 0101 ” が出力される。

20

【 0 1 1 2 】

[終端のゼロラン処理]

なお、ゼロランがラインの最後 (右端) まで連続する場合、例えば、以下のように、さらに短い符号が用いられるようにし、符号量が低減されるようにしてもよい。

【 0 1 1 3 】

図 6 は、ゼロラン符号化の、他の例を示す図である。図 6 に示される例のように、ゼロランが最後まで続く場合、ゼロランの終端は、ラインの最後であることは明らかなので、終端を示す符号 (最大有効桁数 B を示す符号) が、不要になる。

30

【 0 1 1 4 】

換言するに、上述したゼロランを示す符号において、(N - 1) 個の符号 0 が、処理対象ラインの未処理の組数 (現在処理対象の組からライン最後の組までの組数) より多い場合、ゼロランがラインの最後まで続くことが明らかである。

【 0 1 1 5 】

そこで、図 6 の例のように、ゼロランが最後まで続く場合、(0 連長 M の 2 進表現 - 1) の桁数個の符号 0 の後に、符号 1 を並べたものが、このゼロランを示す符号として出力され、その時点でこのラインについての符号化が終了される。

【 0 1 1 6 】

例えば、エントロピ符号化部 2 3 は、ラインの最後まで続く各 0 連長に対して、図 7 に示されるような表 (テーブル情報) を予め記憶しておく。エントロピ符号化部 2 3 は、ゼロランの長さ (0 連長) M を求め、このゼロランがラインの最後まで続くことが判明すると、図 7 のテーブル情報を用いて、その M に対応する符号を選択し、その符号を出力する。

40

【 0 1 1 7 】

このようにすることにより、図 4 の例の場合に比べて、ゼロランを示す符号の符号量を低減させることができる。また、その後の、最大有効桁数 B を示す符号も省略することができる。したがって、エントロピ符号化部 2 3 は、符号化効率をさらに向上させることができる。

【 0 1 1 8 】

50

なお、このようなゼロランを示す符号において、ゼロランがラインの最後まで続くことは、符号0の数により明らかとなる。したがって、ゼロランを示す符号において、例えば、図8の例に示されるように、(0連長Mの2進表現 - 1)の桁数個の符号0の後に続く符号1が省略されるようにしてもよい。

【0119】

例えば、エントロピ符号化部23は、ラインの最後まで続く各0連長に対して、図9に示されるような表(テーブル情報)を予め記憶しておく。エントロピ符号化部23は、ゼロランの長さ(0連長)Mを求め、このゼロランがラインの最後まで続くことが判明すると、図9のテーブル情報を用いて、そのMに対応する符号を選択し、その符号を出力する。

10

【0120】

このようにすることにより、図6の例の場合に比べて、ゼロランを示す符号の符号量を低減させることができる。したがって、エントロピ符号化部23は、符号化効率をさらに向上させることができる。

【0121】

[非パイプライン処理]

ところで、以上のような符号化・復号方法の場合、W個の量子化係数を1組とし、各組の、最大有効桁数の符号化、絶対値の符号化、および、正負を示す符号(サイン)の符号化が、図10に示される例のように、順次行われていた。

【0122】

図10は、符号化処理または復号処理の各処理要素のタイミングチャートである。図10において、時刻の矢印は、時系列を示す。また、 t_0 乃至 t_7 は、時刻を示す。さらに、各四角は、符号化処理の処理要素を示す。

20

【0123】

また、 B_i (以下、 $B(i)$ とも称する)は、W個の量子化係数を1組とするときの、処理対象ラインの先頭から $(i+1)$ 番目の $(i+1)$ 番目に処理される組の最大有効桁数の符号化処理または復号処理を示す。なお、変数 i は、0以上の整数である。つまり、また、 abs_i (以下、 $A(i)$ とも称する)は、W個の量子化係数を1組とするときの、処理対象ラインの左から $(i+1)$ 番目の $(i+1)$ 番目に処理される組の各量子化係数の絶対値の符号化処理または復号処理を示す。さらに、 $sign_i$ (以下、 $S(i)$ とも称する)は、W個の量子化係数を1組とするときの、処理対象ラインの左から $(i+1)$ 番目の $(i+1)$ 番目に処理される組の各量子化係数の符号(サイン)の符号化処理または復号処理を示す。

30

【0124】

さらに、図10において、点線矢印は、符号化データにおける符号化処理の各処理要素の符号の並びの順序を示す。つまり、符号化処理の各処理要素の結果は、各組について、 $B(i)$ 、 $A(i)$ 、 $S(i)$ の順に並べられる。図10の例の場合、符号は、 B_0 、 abs_0 、 $sign_0$ 、 B_1 、 abs_1 、 $sign_1$ 、 B_2 、 abs_2 、 $sign_2$ 、・・・の順に出力される。

【0125】

[非パイプライン処理のゼロランの処理]

つまり、各処理要素の符号は、以下のような順序に並ぶ(図11A)。

40

【0126】

... $B(3)$ $A(3)$ $S(3)$ $B(4)$ $A(4)$ $S(4)$ $B(5)$ $A(5)$ $S(5)$ $B(6)$ $A(6)$ $S(6)$...

【0127】

ここで、 $B(4) = B(5) = 0$ とすると、 $A(4)$ 、 $S(4)$ 、 $A(5)$ 、 $S(5)$ の符号が無くなる(長さ0の符号となる)。長さ0の符号を - で示すと以下の通りになる(図11B)。

【0128】

... $B(3)$ $A(3)$ $S(3)$ $B(4)$ - - $B(5)$ - - $B(6)$ $A(6)$ $S(6)$...

【0129】

この $B(4)$ 、 $B(5)$ 、 $B(6)$ の部分の符号化を、上述したゼロランのルールに沿って行くと、

50

B(4)、B(5)、B(6)がまとめて、diff, minus, more*, enoughという符号に置き換えられた。つまり、B(4)、B(5)、B(6)は連続して並んでいるので、符号化の場合、0連長が確定する周期6 (cycle6)において、上述したB(4)、B(5)、B(6)の符号(diff, minus, more*, enough)が出力された(図11C)。

【0130】

これに対して、復号の場合、B(4)、B(5)、B(6)が始まる周期4 (cycle4)において、B(4)、B(5)、B(6)の符号(diff, minus, more*, enough)が復号された(図11D)。

【0131】

[非パイプライン処理の制約]

復号処理においては、取得した符号化データが順次復号される。つまり、復号処理の各処理要素は、この符号の並び順に行われる。そのため、復号処理の各処理要素を実行するために、以下のような制約が生じる。

【0132】

実行される処理 必要な情報

B(i+1)の頭出し S(i)の頭、S(i)の長さ

B(i+1)の復号 B(i)の復号結果

A(i+1)の頭出し B(i+1)の頭、<B(i+1)の長さ>

A(i+1)の復号 <B(i+1)の復号結果>

S(i+1)の頭出し <A(i+1)の頭>、<B(i+1)の復号結果>

S(i+1)の復号 <A(i+1)復号結果>

【0133】

つまり、絶対値の復号を行うために、符号化データから絶対値の符号化データを得る必要があるが、そのためには、最大有効桁数の符号化データの終端位置を明らかにする必要がある。そのため、絶対値の復号を行う前に、最大有効桁数の符号化データの先頭位置とその符号長を明らかにしなければならない。また、絶対値の符号化データは、最大有効桁数分の絶対値を用いて生成されるので、絶対値の復号には、最大有効桁数の情報が必要になる。

【0134】

また、サイン(正負の符号)の復号を行うために、符号化データからサインの符号化データを得る必要があるが、そのためには、絶対値の符号化データの終端位置を明らかにする必要がある。そのため、サインの復号を行う前に、絶対値の符号化データの先頭位置とその符号長を明らかにしなければならない。また、絶対値が0の場合、サインの符号化データは存在しないので、サインの復号には、各量子化係数の絶対値の情報が必要になる。

【0135】

しかしながら、時刻 $t(i+1)$ からみて、1つ前のタイミング(時刻 t_i)で開始された処理B(i)、A(i)、S(i)の復号が終わった時点においては、上述した必要な情報の内、<>で囲まれた情報が、まだ不明である。

【0136】

つまり、B(i)、A(i)、S(i)の復号が終わった時点で、

- ・B(i+1)の復号を開始する情報は揃っている

- ・A(i+1)の復号を開始するには、B(i+1)の復号が終わるのを待つ必要がある

- ・S(i+1)の復号を開始するには、A(i+1)の復号が終わるのを待つ必要がある

ということが分かる。

【0137】

つまり、復号処理においては、図10の実線矢印に示されるような依存関係が生じる。したがって、B、AおよびSの各処理要素は、逐次に処理する必要があり、並列処理化することが出来なかった。

【0138】

例えば、図10において、abs2の処理を実行するためには、B2の処理結果が必要であるので、時刻 t_2 になり次第、B1,abs1、およびsign1(1つ前の時刻 t_1 に開始される処理

10

20

30

40

50

)の処理結果を用いて処理を開始することができなかった。

【0139】

また、例えば、図10において、sign3の処理を実行するためには、abs3の処理結果が必要であるので、時刻 t_3 になり次第、B2,abs2、およびsign2(1つ前の時刻 t_2 に開始される処理)の処理結果を用いて処理を開始することができなかった。

【0140】

[パイプライン処理1]

そこで、例えば、図12に示されるように、1組に対する復号処理の、各処理要素(B(i)、A(i)、およびS(i))が互いに異なるタイミング(時刻)に実行される(開始される)ようにする。つまり、復号処理の各処理要素(B、AおよびS)を、それぞれパイプライン処理のように実行する。つまり、復号処理の各処理要素(B、AおよびS)を並行して実行することができるようにする。

10

【0141】

図12は、パイプライン処理を行う場合の各処理要素の実行タイミングを示すタイミングチャートの例を説明する図である。

【0142】

図12の例の場合、絶対値の復号処理A(i)の実行タイミングを、最大有効桁数の復号処理B(i)の実行タイミングより1組分遅らせる。また、符号(サイン)の復号処理S(i)の実行タイミングを、最大有効桁数の復号処理B(i)の実行タイミングより2組分遅らせる。

【0143】

20

つまり、例えば、時刻 $t(i+2)$ において、 i 番目の組の符号(サイン)の復号処理S(i)、 $(i+1)$ 番目の組の絶対値の復号処理A(i+1)、 $(i+2)$ 番目の組の最大有効桁数の復号処理B(i+2)が実行され、その後、時刻 $t(i+3)$ において、 $(i+1)$ 番目の組の符号(サイン)の復号処理S(i+1)、 $(i+2)$ 番目の組の絶対値の復号処理A(i+2)、 $(i+3)$ 番目の組の最大有効桁数の復号処理B(i+3)が実行されるようにする。

【0144】

換言するに、符号化データにおいては、互いに同じ時刻に開始される各処理要素B、A、およびSの各符号化処理結果(符号化データ)が、S、A、Bの順に並べられている。

【0145】

このようにすることにより、この場合の、復号処理の各処理要素の実行のための制約は、以下ようになる。

30

【0146】

実行される処理	必要な情報
S(i+1)の頭出し	B(i+2)の頭、B(i+2)の長さ
S(i+1)の復号	A(i+1)の復号結果
A(i+2)の頭出し	S(i+1)の頭、A(i+1)の復号結果から計算できるS(i+1)の長さ
A(i+2)の復号	B(i+2)の復号結果
B(i+3)の頭出し	A(i+2)の頭、B(i+2)の復号結果から導かれるA(i+2)の長さ
B(i+3)の復号	B(i+2)の復号結果

【0147】

40

つまり、この場合、どの処理についても、1つ前の時刻において開始される処理が終了した時点で、必要な情報が全て揃う。したがって、処理対象の組の復号処理のどの処理要素も、現在時刻において開始することができる。

【0148】

つまり、復号処理における各処理要素の依存関係は、図12の実線矢印に示されるような関係となる。したがって、1つの組に対するB、AおよびSの各処理要素間の依存関係が解消されるので、これらを並列処理化することが出来る。したがって、復号処理のスループットを向上させることができる。

【0149】

なお、このように復号処理を行うために、符号化データにおいて、互いに同じ時刻に開

50

始される各処理要素 B、A、および S の各符号化処理結果（符号化データ）が、S、A、B の順に並べられている。処理時間を増大させる並び替え処理等を行わずに、このような並びを実現させるためには、符号化処理も同様の手順で行えばよい。換言するに、符号化処理においてもこのような手順で符号化を行うことにより、復号処理の場合と同様に、符号化処理のスループットを向上させることができる。

【0150】

ハードウェアの場合、図10の例のように B、A、および S を逐次処理する場合でも、各処理要素のために別々の回路を使っていた。そのため、回路規模を増大させずに、上述したような高速化のための並列処理を実現することができる。

【0151】

また、ソフトウェアの場合、B の処理をはじめてから、B に関連する結果が揃うまでの時間、A の処理をはじめてから、A に関連する結果が揃うまでの時間などにある CPU のパイプライン処理（pipeline）の空きを有効利用することが出来る。

【0152】

[パイプライン処理1のゼロランの処理]

このように処理を導入した場合、各処理要素の符号は、以下のように並んでいた（図13A）。

【0153】

... S(2) A(3) B(4) S(3) A(4) B(5) S(4) A(5) B(6) S(5) A(6) B(7) ...

【0154】

B(4) = B(5) = 0 とすると、A(4)、S(4)、A(5)、S(5) の符号は無くなる（長さ0の符号となる）。長さ0の符号を - で示すと以下の通りになる（図13B）。

【0155】

... S(2) A(3) B(4) S(3) - B(5) - - B(6) - A(6) B(7) ...

【0156】

この B(4)、B(5)、B(6) の部分の符号化を、上述したゼロランのルールに沿って行くと、この場合、(4) と B(5) の間に S(3) が存在するので、B(4)、B(5)、B(6) をまとめて、diff, minus, more*, enough という符号に置き換えることが困難になった。

【0157】

例えば、符号化の場合、B(6) で B が 0 でないことが分かってから 0 連長が決定するので、B(6) のタイミングでないと、ゼロランを示す符号が確定しない。そのため、ゼロランを示す符号の位置が S(3) よりも後になってしまう（図13C）。

【0158】

これに対して、復号の場合、S(3) よりも前である B(4) のタイミングで B の値を復号しないと、A(4) をどう扱うべきかが明らかにならない。そのため、ゼロランを示す符号が S(3) よりも前に必要であった（図13D）。このように、符号化の都合にあわせると順次復号を行うことが困難になる恐れがあった。

【0159】

[パイプライン処理2]

なお、パイプライン処理（pipeline）の形は他にも可能である。図14は、パイプライン処理を行う場合のタイミングチャートの、他の例を説明する図である。

【0160】

例えば、図14に示されるように、符号化データにおいて、互いに同じ時刻に開始される各処理要素の符号 B、A、および S が、B、A、S の順に並べられるようにしてもよい。符号化データがこのような並びであっても、各処理要素 B(i+3)、A(i+2)、S(i+1) の実行に必要な情報は、図12の場合と同様に、時刻 t(i+3) において揃っている。したがって、時刻 t(i+3) において、各処理要素 B(i+3)、A(i+2)、S(i+1) を並列に実行することができる。つまり、図12の場合と同様に、符号化処理や復号処理のスループットを向上させることができる。

【0161】

10

20

30

40

50

[パイプライン処理 2 のゼロランの処理]

このように処理を導入した場合、各処理要素の符号は、以下のように並んでいた。

【 0 1 6 2 】

... B(4) A(3) S(2) B(5) A(4) S(3) B(6) A(5) S(4) B(7) A(6) S(5) B(8) A(7) S(6)
...

【 0 1 6 3 】

B(4) = B(5) = 0 とすると、A(4)、S(4)、A(5)、S(5)の符号が無くなる（長さ 0 の符号となる）。長さ 0 の符号を - で示すと以下の通りになる。

【 0 1 6 4 】

... B(4) A(3) S(2) B(5) - S(3) B(6) - - B(7) A(6) - B(8) A(7) S(6)
...

10

【 0 1 6 5 】

このB(4)、B(5)、B(6)の部分の符号化を、上述したゼロランのルールに沿って行くと、この場合、(4)とB(5)の間にS(3)とS(2)が存在するので、B(4)、B(5)、B(6)をまとめて、diff, minus, more*, enoughという符号に置き換えることが困難になった。

【 0 1 6 6 】

例えば、符号化の場合、B(6)でBが0でないことが分かってから 0 連長が決定するので、B(6)のタイミングでないと、ゼロランを示す符号が確定しない。そのため、ゼロランを示す符号の位置がS(3)やS(2)よりも後になってしまう。

【 0 1 6 7 】

20

これに対して、復号の場合、S(3)よりも前であるB(4)のタイミングでBの値を復号しないと、A(4)をどう扱うべきかが明らかにならない。そのため、ゼロランを示す符号がS(3)よりも前に必要であった。このように、この場合も、符号化の都合にあわせると順次復号を行うことが困難になる恐れがあった。

【 0 1 6 8 】

[パイプライン処理に対応したゼロラン処理]

そこで、図 1 5 A に示される例のように、B(4)、B(5)、B(6)の符号を分割してS(3)の前後に並べるようにする。すなわち、diff、minus、並びにmore*をS(3)の前に並べ（配置し）、0 連長符号、並びに、more*、および、enoughをS(3)の後に並べる（配置する）。

【 0 1 6 9 】

30

このようにすることにより、符号化の場合、図 1 5 B に示される例のように、周期 4 (cycle4) において、B(4)を符号化し、周期 5 (cycle5) において、S(3)を符号化し、周期 6 (cycle6) において、B(5)、B(6)を符号化することができる。すなわち、問題なく符号化を行うことができる。

【 0 1 7 0 】

また、復号の場合、図 1 5 C に示される例のように、周期 4 (cycle4) において、B(4)を復号し、周期 5 (cycle5) において、S(3)、B(5)、B(6)をそれぞれ復号することができる。すなわち、問題なく符号化を行うことができる。

【 0 1 7 1 】

40

なお、0 連長符号、並びに、more*、および、enoughを復号する周期 (cycle) は、その他の処理要素の復号との順序を維持していればよい。例えば、図 1 5 の例の場合、図 1 5 D に示されるように、0 連長符号、並びに、more*、および、enoughの復号を、周期 5 (cycle5) において行うようにしても良いし、周期 6 (cycle6) において行うようにしても良い。また、図 1 5 D に示されるように、0 連長符号の復号を周期 5 (cycle5) において行い、more*、および、enoughの復号を周期 6 (cycle6) において行う等、複数の周期に分けてこれらの復号を行うようにしても良い。

【 0 1 7 2 】

[エントロピ符号化部]

以上のように、ゼロランをまとめて符号化するとともに、各処理要素を並列に処理する符号化処理を行う処理部について具体的に説明する。図 1 6 は、エントロピ符号化部 2 3

50

の主な構成例を示すブロック図である。

【 0 1 7 3 】

図 1 6 に示されるように、エントロピ符号化部 2 3 は、ライン判定部 6 1、VLC (Variable Length Coding) 符号化部 6 2、最大有効桁数計算部 6 3、VLC 符号化部 6 4、有効桁抽出部 6 5、VLC 符号化部 6 6、サイン抽出部 6 7、VLC 符号化部 6 8、および符号連結部 6 9 を有する。

【 0 1 7 4 】

さらに、エントロピ符号化部 2 3 は、遅延部 7 1 乃至遅延部 7 5 を有する。

【 0 1 7 5 】

量子化部 2 2 (図 1) から出力された量子化係数は、ライン判定部 6 1 および最大有効桁数計算部 6 3 に供給 (入力) される。また、量子化部 2 2 から出力された量子化係数は、遅延部 7 1 を介して有効桁抽出部 6 5 にも供給される。さらに、量子化部 2 2 から出力された量子化係数は、遅延部 7 3 および遅延部 7 4 を介して、サイン抽出部 6 7 にも供給される。

10

【 0 1 7 6 】

遅延部 7 1 乃至遅延部 7 5 は、例えば、フリップフロップ (FF) 等により構成され、情報の供給を 1 組分遅延させる。つまり、遅延部 7 1 乃至遅延部 7 5 は、供給される情報を保持し、次の組の処理タイミングにおいて、その保持している情報を後段の処理部に供給する。例えば、遅延部 7 1 乃至遅延部 7 5 は、時刻 t_i に行われた処理の結果として供給される情報を、一時的に保持し、時刻 $t(i+1)$ において、後段の処理部に供給する。つまり、その後段の処理部においては、遅延部 7 1 乃至遅延部 7 5 から供給された情報を、時刻 $t(i+1)$ に行う処理に利用する。

20

【 0 1 7 7 】

より具体的には、遅延部 7 1 は、量子化部 2 2 から出力された量子化係数や、最大有効桁数計算部 6 3 から供給される最大有効桁数の計算結果を、次の組の処理タイミングにおいて、有効桁抽出部 6 5 に供給する。また、遅延部 7 2 は、最大有効桁数計算部 6 3 から供給される最大有効桁数の計算結果を、次の組の処理タイミングにおいて、有効桁抽出部 6 5 に供給する。

【 0 1 7 8 】

遅延部 7 3 は、量子化部 2 2 から出力された量子化係数や、最大有効桁数計算部 6 3 から供給される最大有効桁数の計算結果を、次の組の処理タイミングにおいて、遅延部 7 4 に供給する。遅延部 7 4 は、遅延部 7 3 から供給されるこれらの情報を、次の組の処理タイミングにおいて、サイン抽出部 6 7 に供給する。

30

【 0 1 7 9 】

また、遅延部 7 5 は、有効桁抽出部 6 5 から供給される量子化係数の有効桁 (のデータ) を、次の組の処理タイミングにおいて、サイン抽出部 6 7 に供給する。

【 0 1 8 0 】

ライン判定部 6 1 は、量子化部 2 2 から入力された、これから符号化する 1 つのラインの量子化係数が全て 0 であるか否かを判定し、その判定の結果を示す情報を VLC 符号化部 6 2 に供給する。つまり、ライン判定部 6 1 は、量子化係数の入力タイミングにおいて、処理を行う。

40

【 0 1 8 1 】

VLC 符号化部 6 2 は、ライン判定部 6 1 からの判定の結果を示す情報に基づいて、符号化するラインの量子化係数が全て 0 であるか否かを示す符号を符号連結部 6 9 に出力する。つまり、VLC 符号化部 6 2 は、量子化係数の入力タイミングにおいて、処理を行う。

【 0 1 8 2 】

最大有効桁数計算部 6 3 は、量子化部 2 2 から入力された、連続する W 個の量子化係数の最大有効桁数を計算し、その計算の結果を示す情報を VLC 符号化部 6 4 に供給する。つまり、最大有効桁数計算部 6 3 は、量子化係数の入力タイミングにおいて、処理を行う。なお、最大有効桁数計算部 6 3 は、最大有効桁数の計算結果を示す情報を、遅延部 7 1 お

50

よび遅延部 7 2 を介して、有効桁抽出部 6 5 にも供給する。

【 0 1 8 3 】

VLC符号化部 6 4 は、最大有効桁数計算部 6 3 からの計算の結果を示す情報に基づいて、W個の量子化係数の最大有効桁数を示す符号を符号連結部 6 9 に供給する。また、VLC符号化部 6 4 は、ゼロランが存在する場合、最大有効桁数計算部 6 3 からの計算の結果を示す情報に基づいて、ゼロランを示す符号を符号連結部 6 9 に供給する。つまり、VLC符号化部 6 4 は、量子化係数の入力タイミングにおいて、処理を行う。

【 0 1 8 4 】

有効桁抽出部 6 5 は、遅延部 7 1 および遅延部 7 2 を介して最大有効桁数計算部 6 3 から供給された最大有効桁数の計算結果を示す情報に基づいて、遅延部 7 1 を介して量子化部 2 2 から供給されたW個の量子化係数の有効桁を抽出し、抽出した量子化係数の有効桁（のデータ）をVLC符号化部 6 6 に供給する。つまり、有効桁抽出部 6 5 は、量子化係数の入力タイミングから 1 組分遅延して、処理を行う。なお、有効桁抽出部 6 5 は、抽出した量子化係数の有効桁（のデータ）を、遅延部 7 5 を介して、サイン抽出部 6 7 にも供給する。

10

【 0 1 8 5 】

VLC符号化部 6 6 は、有効桁抽出部 6 5 からの量子化係数の有効桁に基づいて、これらの量子化係数の絶対値を符号化し、これにより得られた量子化係数の絶対値を示す符号を符号連結部 6 9 に供給する。つまり、VLC符号化部 6 6 は、量子化係数の入力タイミングから 1 組分遅延して、処理を行う。

20

【 0 1 8 6 】

サイン抽出部 6 7 は、遅延部 7 5 を介して有効桁抽出部 6 5 から供給された量子化係数の有効桁に基づいて、遅延部 7 3 および遅延部 7 4 を介して量子化部 2 2 から供給された量子化係数のサインを抽出して、抽出したサイン（のデータ）をVLC符号化部 6 8 に供給する。つまり、サイン抽出部 6 7 は、量子化係数の入力タイミングから 2 組分遅延して、処理を行う。

【 0 1 8 7 】

VLC符号化部 6 8 は、サイン抽出部 6 7 からのサイン（のデータ）を符号化し、これにより得られる量子化係数のサインを示す符号を符号連結部 6 9 に供給する。つまり、VLC符号化部 6 8 は、量子化係数の入力タイミングから 2 組分遅延して、処理を行う。

30

【 0 1 8 8 】

符号連結部 6 9 は、VLC符号化部 6 2、VLC符号化部 6 4、VLC符号化部 6 6、およびVLC符号化部 6 8 のそれぞれから供給された、ラインの量子化係数が全て 0 であるか否かを示す符号、最大有効桁数を示す符号、量子化係数の絶対値を示す符号、並びに、量子化係数のサインを示す符号のそれぞれを連結し、符号化された画像（データ）として出力する。

【 0 1 8 9 】

[VLC符号化部]

図 1 7 は、VLC符号化部 6 4 の主な構成例を示すブロック図である。

【 0 1 9 0 】

図 1 7 に示されるように、VLC符号化部 6 4 は、最大有効桁数変化符号化部 8 1、最大有効桁数変化量符号化部 8 2、および 0 連長符号化部 8 3 を有する。

40

【 0 1 9 1 】

最大有効桁数変化符号化部 8 1 は、最大有効桁数 B が、前の組から変化するか否かを検出し、その変化の有無を示す符号を、最大有効桁数を示す符号として、符号連結部 6 9 に供給する。

【 0 1 9 2 】

最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B が、前の組からの変化する場合、その変化量を求め、その変化量を示す符号を、最大有効桁数を示す符号として、符号連結部 6 9 に供給する。

【 0 1 9 3 】

50

0 連長符号化部 8 3 は、ゼロランが存在する場合、そのゼロランの長さ (0 連長) を求め、そのゼロランの長さ (0 連長) を示す符号を、ゼロランを示す符号として、符号連結部 6 9 に供給する。

【 0 1 9 4 】

なお、最大有効桁数変化符号化部 8 1 および最大有効桁数変化量符号化部 8 2 は、ゼロラン先頭の組の最大有効桁数 B の符号化を行う周期 (cycle) において、最大有効桁数 B から最大有効桁数 B = 0 への変化を示す符号 (diff, minus, more*) を出力する。

【 0 1 9 5 】

その後、0 連長符号化部 8 3 は、ゼロランの長さ (0 連長) を計測し、ゼロラン終端の次の組 (最大有効桁数 B = 0 となる組) の最大有効桁数 B の符号化を行う周期 (cycle) において、0 連長符号 (ゼロランの長さを示す符号) を出力する。

10

【 0 1 9 6 】

また、その 0 連長符号と同周期 (cycle) において、最大有効桁数変化符号化部 8 1 および最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B = 0 から最大有効桁数 B = 0 への変化を示す符号 (more*, enough) を出力する。

【 0 1 9 7 】

[パイプライン処理に対応したゼロラン処理]

つまり、新たに B が 0 になるとき、そのタイミングで diff, minus, more* の符号により B が 0 になることを符号化する。(1)

【 0 1 9 8 】

0 連長が確定した時点で、ゼロラン符号により 0 連長を符号化し、ライン最後まで 0 ではない場合、続けて more*, enough の符号により、次の 0 でない B の値を符号化する。(2)

20

【 0 1 9 9 】

符号化する係数情報を見て確定した値を符号化しているため、この符号化は可能である。また、図 1 3 を参照して説明した場合の、S(3) は、(1) と同じタイミングで符号化され、(1) の後ろに接続される。そのため、(1) の符号と (2) の符号の間に S(3) が置かれることになる。これにより、復号も可能になる。

【 0 2 0 0 】

[符号化処理の流れ]

次に、以上のような画像符号化装置 1 1 の各部が実行する処理の具体的な流れについて説明する。

30

【 0 2 0 1 】

最初に、図 1 8 のフローチャートを参照して、画像符号化装置 1 1 (図 1) による、符号化処理の流れの例を説明する。この符号化処理は、ウェーブレット変換部 2 1 に、符号化する画像 (データ) が入力されると開始される。

【 0 2 0 2 】

ステップ S 1 1 において、ウェーブレット変換部 2 1 は、入力された画像にウェーブレット変換を施して、入力された画像を複数のサブバンドに分解し、各サブバンドのウェーブレット係数を量子化部 2 2 に供給する。

40

【 0 2 0 3 】

ステップ S 1 2 において、量子化部 2 2 は、ウェーブレット変換部 2 1 から供給されたウェーブレット係数を量子化し、その結果得られた量子化係数をエントロピ符号化部 2 3 に供給する。これにより、エントロピ符号化部 2 3 には、例えば、図 3 を参照して説明したビットプレーン表現された、サブバンドの各位置の量子化係数が入力される。

【 0 2 0 4 】

ステップ S 1 3 において、エントロピ符号化部 2 3 は、エントロピ符号化処理を行い、符号化処理を終了する。なお、エントロピ符号化処理の詳細は後述するが、エントロピ符号化部 2 3 は、エントロピ符号化処理において、図 3 を参照して説明したように、量子化部 2 2 から供給された量子化係数を所定の数 (W 個) ずつ符号化し、符号化するラインの

50

量子化係数が全て0であるか否かを示す符号、量子化係数の最大有効桁数を示す符号、ゼロランを示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号を、符号化された画像（符号化データ）として出力する。

【0205】

その際、エントロピ符号化部23は、エントロピ符号化処理の各処理要素を、図15を参照して説明したようなタイミングで、互いに並列に実行する。これによりエントロピ符号化部23は、エントロピ符号化処理のスループットを向上させることができる。また、符号化するラインに値が0の量子化係数が連続するゼロランをまとめて符号化することができるので、エントロピ符号化部23は、符号化効率を向上させることができる。

【0206】

このようにして、画像符号化装置11は、入力された画像を符号化して出力する。

【0207】

[エントロピ符号化処理の流れ]

次に、図19のフローチャートを参照して、図18のステップS13の処理に対応するエントロピ符号化処理について説明する。

【0208】

図18のステップS12において、量子化部22から出力された量子化係数は、エントロピ符号化部23のライン判定部61、最大有効桁数計算部63、（遅延部71を介して）有効桁抽出部65、および（遅延部73および遅延部74を介して）サイン抽出部67に供給（入力）される。

【0209】

ステップS41において、ライン判定部61は、これから符号化するサブバンドのラインを示す変数 y を $y=0$ として、これを記憶する。

【0210】

例えば、図2に示したサブバンドの量子化係数を符号化する場合、ライン判定部61は、そのサブバンドのライン（ラインL1乃至ラインL6）を示す変数 y を、 $y=0$ とする。なお、ここで、変数 y により示されるライン y は、サブバンドのライン上の各位置 (x, y) の y 座標が y であるラインを示している。したがって、例えば、ライン判定部61が記憶している変数 y が $y=0$ である場合、その変数により示されるラインは、ライン上の各位置の y 座標が0であるラインL1となる。

【0211】

ステップS42において、最大有効桁数計算部63は、ライン判定部61が記憶している変数 y により示されるライン y よりも1つ前のライン $(y-1)$ 上の最初に入力される W 個の量子化係数の最大有効桁数を示す変数 B_{init} を $B_{init}=0$ として、これを記憶する。

【0212】

例えば、ライン $(y-1)$ が、図2に示したラインL1である場合、ライン $(y-1)$ 上の最初に入力される W 個の量子化係数の最大有効桁数を示す変数 B_{init} の値は、ラインL1の図中、左端の位置から W 個の量子化係数、すなわち、位置 $(0, 0)$ 、 $(1, 0)$ 、 \dots 、 $(w-1, 0)$ の W 個の量子化係数の最大有効桁数となる。また、ライン判定部61が記憶している変数 y が $y=0$ である場合、ライン $(y-1)$ は存在しないので、変数 B_{init} の値は、 $B_{init}=0$ とされる。

【0213】

ステップS43において、ライン判定部61は、記憶している変数 y により示されるライン y の量子化係数（の絶対値）が全て0であるか否かを判定する。例えば、ライン y が、図2に示したラインL1である場合、ライン判定部61は、ラインL1上の位置 (x, y) の量子化係数が全て0であるとき、量子化係数が全て0であると判定する。

【0214】

ステップS43において、量子化係数が全て0であると判定した場合、ライン判定部61は、量子化係数が全て0である旨の情報を生成してそれをVLC符号化部62および最大有効桁数計算部63に供給し、処理をステップS44に進める。

10

20

30

40

50

【0215】

ステップS44において、VLC符号化部62は、ライン判定部61からの量子化係数が全て0である旨の情報に基づいて、符号化するラインの量子化係数が全て0であることを示す符号0を符号連結部69に出力（供給）する。符号連結部69は、VLC符号化部62から供給された符号0を、ラインyの量子化係数の符号化の結果得られた符号として、そのまま出力する。

【0216】

ステップS45において、最大有効桁数計算部63は、ライン判定部61からの量子化係数が全て0である旨の情報に基づいて、記憶している変数Binitの値を $Binit = 0$ とし、変数Binitを更新する。

10

【0217】

ステップS46において、ライン判定部61は、符号化しているサブバンドのラインのうち、未処理のラインがあるか否かを判定する。すなわち、ライン判定部61は、符号化しているサブバンドの全てのラインの量子化係数を符号化したか否かを判定する。例えば、図2に示したサブバンドの量子化係数を符号化している場合、ラインL1乃至ラインL6上の全ての位置の量子化係数が符号化されたとき、ライン判定部61は、未処理のラインが存在しないと判定する。

【0218】

そのステップS46において、未処理のラインがあると判定した場合、ライン判定部61は、次のライン、すなわち、ライン $(y+1)$ 上の各位置の量子化係数を符号化するので、処理をステップS47に進める。

20

【0219】

ステップS47において、ライン判定部61は、記憶しているラインを示す変数yをインクリメントして $y = y + 1$ とし、処理をステップS43に戻し、上述したそれ以降の処理を再び実行させる。

【0220】

これに対して、ステップS46において、未処理のラインが無いと判定した場合、ライン判定部61は、サブバンドを構成する全てのラインについて量子化係数を符号化したので、エントロピ符号化処理を終了し、処理を図18のステップS13に戻し、符号化処理を終了させる。

30

【0221】

また、図19のステップS43において、ラインyの量子化係数が全て0でない（0でない量子化係数が存在する）と判定した場合、ライン判定部61は、量子化係数が全て0でない（0でない量子化係数が存在する）旨の情報を生成してそれをVLC符号化部62および最大有効桁数計算部63に供給し、ステップS48に処理を進める。

【0222】

ステップS48において、VLC符号化部62は、ライン判定部61からの量子化係数が全て0でない旨の情報に基づいて、符号化するラインの量子化係数が全て0でないことを示す符号1を符号連結部69に出力（供給）する。符号連結部69は、VLC符号化部62から供給された符号1を、そのまま出力する。

40

【0223】

ステップS49において、エントロピ符号化部23はW個組符号化処理を行う。なお、W個組符号化処理の詳細は後述するが、W個組符号化処理において、エントロピ符号化部23は、ライン判定部61が記憶している変数yにより示されるラインy上の量子化係数を、連続するW個組毎に符号化する。

【0224】

ここで、ライン判定部61が記憶している変数yおよび、最大有効桁数計算部63が記憶している変数xにより特定されるラインy上の位置を (x, y) とすると、ラインy上のW個の連続する位置は、ラインy上の連続する位置 (x, y) 、 $(x+1, y)$ 、 \dots 、 $(x+W-1, y)$ とされる。すなわち、W個組符号化処理において、エントロピ符号化部23は、

50

位置 (x, y) 、 $(x+1, y)$ 、 \dots 、 $(x+w-1, y)$ のそれぞれの量子化係数を符号化する。

【0225】

W個組符号化処理が終了すると、エントロピ符号化部23は、処理をステップS46に戻し、それ以降の処理を実行させる。

【0226】

このようにして、エントロピ符号化部23は、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に符号化する。

【0227】

このように、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に符号化することによって、入力された量子化係数を入力された順番に処理することができ、量子化係数の符号化により生じる遅延をより少なくすることができる。

10

【0228】

[W個組符号化処理の流れ]

次に、図20のフローチャートを参照して、図19のステップS49の処理に対応するW個組符号化処理について説明する。

【0229】

W個組符号化処理が開始されると、エントロピ符号化部23は、ステップS61において、 i 、 B 、および \max 等の各種変数を初期化する。例えば、エントロピ符号化部23は、変数 i と B の値をそれぞれ0に設定し ($i = 0$ 、 $B = 0$)、変数 \max の値を、処理対象である当該ライン上の係数W個組の数とする ($\max =$ 係数W個組の数)。

20

【0230】

ステップS62において、サイン抽出部67およびVLC符号化部68は、 $(i - 1)$ 番目の係数W個組が存在する場合、その各係数について、絶対値が0でない各量子化係数のサインを示す符号を並べたものを、当該組のサインを示す符号として生成し、符号連結部69に出力(供給)する。符号連結部69は、VLC符号化部68から供給された符号を、ライン y の $(i - 1)$ 番目の組の絶対値を示す符号として、そのまま出力する。なお、 $(i - 1)$ 番目の係数W個組が存在しない場合、この処理は省略(スキップされる)。

【0231】

ステップS63において、有効桁抽出部65およびVLC符号化部66は、 i 番目の係数W個組が存在する場合、その各係数について、前回のステップS64の処理により求められた最大有効桁数 B ビット分の絶対値を抽出し、それらを並べたものを、当該組の絶対値を示す符号として生成し、符号連結部69に出力(供給)する。符号連結部69は、VLC符号化部66から供給された符号を、ライン y の i 番目の組の絶対値を示す符号として、そのまま出力する。なお、 i 番目の係数W個組が存在しない場合、この処理は省略(スキップされる)。

30

【0232】

ステップS64において、最大有効桁数計算部63およびVLC符号化部64は、最大有効桁数処理を行い、最大有効桁数を示す符号を生成し、それを符号連結部69に出力(供給)する。符号連結部69は、VLC符号化部64から供給された符号を、ライン y の $(i + 1)$ 番目の組の最大有効桁数を示す符号として、そのまま出力する。

40

【0233】

ステップS64の処理が終了すると、最大有効桁数計算部63およびVLC符号化部64は、処理をステップS65に進める。

【0234】

ステップS65において、エントロピ符号化部23は、変数 i ($\max + 1$)であるかを判定する。すなわち、エントロピ符号化部23は、処理対象のラインの全ての量子化係数の組を処理したか否かを判定する。

【0235】

変数 i ($\max + 1$)であり、処理対象のライン上に未処理の組が存在すると判定され

50

た場合、エントロピ符号化部 2 3 は、処理をステップ S 6 6 に進める。

【 0 2 3 6 】

ステップ S 6 6 において、エントロピ符号化部 2 3 は、変数 i をインクリメント（変数 $i = i + 1$ ）し、処理をステップ S 6 2 に戻す。

【 0 2 3 7 】

つまり、ステップ S 6 2 の処理は、図 1 6 の遅延部 7 3 および遅延部 7 4 により、ステップ S 6 4 の処理に比べて、2 周期（2cycle）分遅延されている。したがって、ステップ S 6 2 においては、ステップ S 6 4 の 2 組前の処理対象の組が処理対象とされる。

【 0 2 3 8 】

また、ステップ S 6 3 の処理は、図 1 6 の遅延部 7 1 により、ステップ S 6 4 の処理に比べて、1 周期（1cycle）分遅延されている。したがって、ステップ S 6 3 においては、ステップ S 6 4 の 1 組前の処理対象の組が処理対象とされる。

10

【 0 2 3 9 】

変数 i は、0 以上の整数である。したがって、例えば、 $i = 0$ の場合、ステップ S 6 4 の処理のみが行われる。また、例えば、 $i = 1$ の場合、ステップ S 6 3 およびステップ S 6 4 の処理が行われる。 $2 \leq i < \max$ の場合、ステップ S 6 2 乃至ステップ S 6 4 の処理が行われる。 $i = \max$ の場合、ステップ S 6 2 およびステップ S 6 3 の処理が行われる。 $i = \max + 1$ の場合、ステップ S 6 2 の処理のみが行われる。

【 0 2 4 0 】

そして、ステップ S 6 5 において、 $i > (\max + 1)$ であると判定した場合、エントロピ符号化部 2 3 は、処理をステップ S 6 7 に進める。

20

【 0 2 4 1 】

ステップ S 6 7 において、VLC符号化部 6 4 は、ステップ S 6 4 の最大有効桁数処理において計測したゼロランの長さ（0 連長）を示す変数 $count$ が 0 でない場合、ライン終端に達するゼロランを示す符号として、 $(count - 1)$ の桁数個の符号 0 を生成し、それを符号連結部 6 9 に出力（供給）する。符号連結部 6 9 は、VLC符号化部 6 4 から供給された符号を、ライン y のライン終端に達するゼロランを示す符号として、そのまま出力する。

【 0 2 4 2 】

なお、変数 $count$ の値が 0 である場合、この処理は省略（スキップ）される。

30

【 0 2 4 3 】

ステップ S 6 7 の処理が終了すると、VLC符号化部 6 4 は、 W 個組符号化処理を終了し、処理を図 1 9 に戻す。

【 0 2 4 4 】

[最大有効桁数処理の流れ]

次に、図 2 1 および図 2 2 のフローチャートを参照して、図 2 0 のステップ S 6 4 において実行される最大有効桁数処理の流れの例を説明する。

【 0 2 4 5 】

最大有効桁数処理が開始されると、ステップ S 7 1 において、最大有効桁数計算部 6 3 は、 $(i + 1)$ 番目の係数 W 個組のうち、絶対値が最も大きい量子化係数の有効桁数を、これから符号化する W 個の量子化係数の最大有効桁数を示す変数 B_{new} の値として、この変数 B_{new} を記憶する。

40

【 0 2 4 6 】

また、最大有効桁数計算部 6 3 は、求められた W 個の量子化係数の最大有効桁数、すなわち、変数 B_{new} の値を VLC符号化部 6 4 に供給する。また、最大有効桁数計算部 6 3 は、その変数 B_{new} の値を、遅延部 7 1 若しくは遅延部 7 2 を介して、有効桁抽出部 6 5 に供給する。

【 0 2 4 7 】

例えば、 W 個の連続する位置の量子化係数のそれぞれが、図 3 に示した量子化係数 “ -0101 ”、“ +0011 ”、“ -0110 ”、および “ +0010 ” である場合、これらの量子化係数のうち

50

、絶対値が最大の量子化係数は“-0110”であり、その有効桁数は“-0110”の最上位にある1の桁である“3”であるので、変数Bnewの値は3とされる。

【0248】

ステップS72において、VLC符号化部64は、ステップS71の処理結果に基づいて、(i+1)番目の係数W個組が存在するか否かを判定する。存在しないと判定した場合、VLC符号化部64は、最大有効桁数処理を終了し、処理を図20に戻す。例えば、imaxの場合、(i+1)番目の係数W個組は、存在しない。このような場合、ステップS73乃至ステップS82の処理は省略(スキップ)される。

【0249】

(i+1)番目の係数W個組が存在すると判定した場合、VLC符号化部64は、処理をステップS73に進める。

10

【0250】

ステップS73において、0連長符号化部83は、変数count=0であるか否かを判定する。ゼロラン中でなく、count=0であると判定した場合、0連長符号化部83は、処理をステップS74に進める。

【0251】

ステップS74において、最大有効桁数変化符号化部81は、Bnew=Bであるか否かを判定する。すなわち、最大有効桁数変化符号化部81は、記憶している、前回符号化されたW個の量子化係数の最大有効桁数を示す変数Bの値が、最大有効桁数計算部63から供給された、これから符号化するW個の量子化係数の最大有効桁数を示す変数Bnewの値と同じであるか否かを判定する。

20

【0252】

Bnew=Bであると判定した場合、最大有効桁数変化符号化部81は、処理をステップS75に進め、これから符号化するW個の量子化係数の最大有効桁数を示す符号として、最大有効桁数が変化していないことを示す符号0を符号連結部69に出力する。そして、最大有効桁数を示す符号0を出力すると、最大有効桁数変化符号化部81は、ステップS76乃至ステップS80の各処理を省略(スキップ)し、ステップS81に処理を進める。

【0253】

これに対して、ステップS74において、Bnew=Bでないとして判定した場合、最大有効桁数変化符号化部81は、処理をステップS76に進め、(最大有効桁数が変化したので、)

30

最大有効桁数が変化したことを示す符号1を符号連結部69に出力する。

【0254】

ステップS77において、最大有効桁数変化符号化部81は、Bnew>Bであるか否かを判定する。

【0255】

処理対象の量子化係数の組(今の組)の最大有効桁数Bnewが、前回処理対象であった量子化係数の組(前の組)の最大有効桁数Bよりも大きい(最大有効桁数が増加した)と判定した場合、最大有効桁数変化符号化部81は、処理をステップS78に進める。

【0256】

ステップS78において、最大有効桁数変化符号化部81は、最大有効桁数が増加したことを示す符号0を、最大有効桁数を示す符号として、符号連結部69に出力する。続いて、最大有効桁数変化量符号化部82は、最大有効桁数の変化量(増分)を示す(Bnew-B-1)個の符号0を、最大有効桁数を示す符号として、符号連結部69に出力する。そして、最大有効桁数変化量符号化部82は、処理をステップS80に進める。

40

【0257】

また、ステップS77において、今の組の最大有効桁数Bnewが、前の組の最大有効桁数Bよりも小さい(最大有効桁数が減少した)と判定した場合、最大有効桁数変化符号化部81は、処理をステップS79に進める。

【0258】

ステップS79において、最大有効桁数変化符号化部81は、最大有効桁数が減少した

50

ことを示す符号 1 を、最大有効桁数を示す符号として、符号連結部 6 9 に出力する。続いて、最大有効桁数変化量符号化部 8 2 は、最大有効桁数の変化量（減分）を示す（ $B - B_{new} - 1$ ）個の符号 0 を、最大有効桁数を示す符号として、符号連結部 6 9 に出力する。そして、最大有効桁数変化量符号化部 8 2 は、処理をステップ S 8 0 に進める。

【 0 2 5 9 】

ステップ S 8 0 において、最大有効桁数変化量符号化部 8 2 は、今の組の最大有効桁数 B_{new} が 0 でない場合、変動の終端（enough）を示す符号 1 を、符号連結部 6 9 に出力する。つまり、ここでは、今の組の最大有効桁数 B_{new} が 0 である場合、この符号 1 の出力が省略（スキップ）されるものとしている。この変動の終端（enough）を示す符号 1 の出力を省略しない場合、最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B の変化量（減分）を示す（ $B - B_{new} - 1$ ）個の符号 0 に続けて、その変動の終端（enough）を示す符号 1 を符号連結部 6 9 に出力すればよい。

10

【 0 2 6 0 】

そして、最大有効桁数変化量符号化部 8 2 は、処理をステップ S 8 1 に進める。

【 0 2 6 1 】

ステップ S 8 1 において、0 連長符号化部 8 3 は、 B_{new} が 0 の場合、すなわち、ゼロランが開始された場合、変数 $count$ の値を 0 から 1 に変更する。そして、0 連長符号化部 8 3 は、処理をステップ S 8 2 に進める。

【 0 2 6 2 】

ステップ S 8 2 において、最大有効桁数計算部 6 3 は、変数 B に変数 B_{new} の値をセットする（ $B = B_{new}$ とする）。

20

【 0 2 6 3 】

ステップ S 8 2 の処理が終了すると、最大有効桁数計算部 6 3 は、最大有効桁数処理を終了し、処理を図 2 0 に戻す。

【 0 2 6 4 】

また、図 2 1 のステップ S 7 3 において、ゼロラン中であり、 $count = 0$ でないと判定した場合、0 連長符号化部 8 3 は、処理を図 2 2 のステップ S 9 1 に進める。

【 0 2 6 5 】

図 2 2 のステップ S 9 1 において、0 連長符号化部 8 3 は、 $B_{new} = 0$ であるか否かを判定する。 $B_{new} = 0$ であり、今の組でもゼロランが終了しないと判定された場合、0 連長符号化部 8 3 は、処理をステップ S 9 2 に進める。

30

【 0 2 6 6 】

ステップ S 9 2 において、0 連長符号化部 8 3 は、ゼロランの組数のカウント値である変数 $count$ をインクリメント（変数 $count = count + 1$ ）し、処理を図 2 1 のステップ S 8 2 に戻し、それ以降の処理を実行させる。

【 0 2 6 7 】

また、図 2 2 のステップ S 9 1 において、 $B_{new} = 0$ でなく、今の組でもゼロランが終了すると判定された場合、0 連長符号化部 8 3 は、処理をステップ S 9 3 に進める。

【 0 2 6 8 】

ステップ S 9 3 において、0 連長符号化部 8 3 は、例えば図 5 のテーブルの一番右の列に示されるような、（変数 $count$ の桁数 - 1）個の符号 0 に、変数 $count$ の 2 進表現を続けたものを、ゼロランを示す符号として符号連結部 6 9 に出力する。

40

【 0 2 6 9 】

ステップ S 9 4 において、最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B の増分を示す（ $B_{new} - 1$ ）個の符号 0 を符号連結部 6 9 に出力する。なお、ここでは、変動の終端（enough）を示す符号 1 の出力は省略（スキップ）されている。この変動の終端（enough）を示す符号 1 の出力を省略しない場合、最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B の増分を示す（ $B_{new} - 1$ ）個の符号 0 に続けて、その変動の終端（enough）を示す符号 1 を符号連結部 6 9 に出力すればよい。

【 0 2 7 0 】

50

ステップS 9 5において、0 連長符号化部 8 3は、変数countの値を0とする。ステップS 9 5の処理が終了すると、0 連長符号化部 8 3は、処理を図 2 1のステップS 8 2に戻し、それ以降の処理を実行させる。

【0 2 7 1】

このようにして、エントロピ符号化部 2 3は、サブバンドの量子化係数を、予め定められた所定の数ずつまとめて符号化し、量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号を出力する。

【0 2 7 2】

このようにすることで、例えば、JPEG2000方式により画像を符号化する場合とは異なり、複数のコーディングパスに基づいて、ビットプレーンごとに複数の処理を行う必要がなく、また、可変長符号化を行うので、符号化の処理量を大幅に低減することができる。これにより、画像符号化装置 1 1は、より容易に画像の符号化を行うことができ、高解像度の画像をリアルタイムで符号化するための符号化装置を安価で実現することができる。

10

【0 2 7 3】

さらに、画像符号化装置 1 1は、符号の長さを明示的に符号化する必要がないため、符号量を少なくすることができ、符号の長さの情報を管理する必要もない。

【0 2 7 4】

また、上述したように、連続する、値が0の量子化係数のみにより構成される組を、まとめてゼロランとして符号化するので、画像符号化装置 1 1は、さらに、0が多い係数データの符号化の符号化効率を向上させることができる。

20

【0 2 7 5】

さらに、そのゼロランの符号化を行いながら、1つの組に対する符号化処理の各処理要素を、互いに異なるタイミングで実行させるようにすることにより、画像符号化装置 1 1は、符号化効率を向上させながら、符号化処理のスループットの向上を実現することができる。その結果、画像符号化装置 1 1は、より高速に符号化処理を行うことができる。

【0 2 7 6】

なお、以上においては、W個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数を、最大有効桁数を示す変数Bnewの値とすると説明したが、変数Bnewの値は、W個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数以上の値であればよい。変数Bnewの値が大きくなると、量子化係数の絶対値を示す符号の符号量が多くなるが、変数Bnewの値を、絶対値の最も大きい量子化係数の有効桁数以上の値とすることで、量子化係数の最大有効桁数を示す符号の符号量を少なくすることができる。

30

【0 2 7 7】

なお、符号化処理の各処理要素の符号化データを、図 1 4の例のように並べる場合、図 2 0のフローチャートにおいて、ステップS 6 2乃至ステップS 6 4の各処理の実行順を、ステップS 6 4、ステップS 6 3、ステップS 6 2の順に入れ替えればよい。

【0 2 7 8】

< 2 . 第 2 の実施の形態 >

[画像復号装置]

図 2 3は、画像復号装置の主な構成例を示すブロック図である。図 2 3に示される画像復号装置 1 1 1は、画像符号化装置 1 1に対応する画像処理装置である。すなわち、画像復号装置 1 1 1は、画像符号化装置 1 1に入力される画像データを符号化して出力した符号化データを、画像符号化装置 1 1の符号化処理に対応する方法で復号し、復号画像データを出力する。

40

【0 2 7 9】

図 2 3に示されるように、画像復号装置 1 1 1は、エントロピ復号部 1 2 1、逆量子化部 1 2 2、およびウェーブレット逆変換部 1 2 3を有する。

【0 2 8 0】

エントロピ復号部 1 2 1には、符号化された画像(データ)が入力される。エントロピ復号部 1 2 1は、入力された符号化された画像としての符号をエントロピ復号し、これに

50

より得られた量子化係数を逆量子化部 1 2 2 に供給する。

【 0 2 8 1 】

逆量子化部 1 2 2 は、エントロピ復号部 1 2 1 から供給された量子化係数を逆量子化し、逆量子化により得られた各サブバンドのウェーブレット係数をウェーブレット逆変換部 1 2 3 に供給する。

【 0 2 8 2 】

ウェーブレット逆変換部 1 2 3 は、逆量子化部 1 2 2 から供給された、各サブバンドのウェーブレット係数にウェーブレット逆変換を施し、その結果得られた画像を、復号された画像として出力する。

【 0 2 8 3 】

[エントロピ復号部]

図 2 4 は、エントロピ復号部 1 2 1 の主な構成例を示すブロック図である。

【 0 2 8 4 】

図 2 4 に示されるように、エントロピ復号部 1 2 1 は、符号分割部 1 5 1、ライン判定部 1 5 2、発生部 1 5 3、VLC復号部 1 5 4、VLC復号部 1 5 5、VLC復号部 1 5 6、量子化係数合成部 1 5 7、および切替部 1 5 8 を有する。

【 0 2 8 5 】

また、エントロピ復号部 1 2 1 は、遅延部 1 6 1 および遅延部 1 6 2 を有する。

【 0 2 8 6 】

遅延部 1 6 1 および遅延部 1 6 2 は、例えば、フリップフロップ (FF) 等により構成され、情報の供給を 1 組分遅延させる。つまり、遅延部 1 6 1 および遅延部 1 6 2 は、供給される情報を保持し、次の組の処理タイミングにおいて、その保持している情報を後段の処理部に供給する。例えば、遅延部 1 6 1 および遅延部 1 6 2 は、時刻 t_i に行われた処理の結果として供給される情報を、一時的に保持し、時刻 $t(i+1)$ において、後段の処理部に供給する。つまり、その後段の処理部においては、遅延部 1 6 1 および遅延部 1 6 2 から供給された情報を、時刻 $t(i+1)$ に行う処理に利用する。

【 0 2 8 7 】

より具体的には、遅延部 1 6 1 は、VLC復号部 1 5 4 から供給される最大有効桁数を示す情報を、次の組の処理タイミングにおいて、VLC復号部 1 5 5 および量子化係数合成部 1 5 7 に供給する。

【 0 2 8 8 】

また、遅延部 1 6 2 は、VLC復号部 1 5 5 から供給される量子化係数の絶対値を示す符号の復号の結果を示す情報を、次の組の処理タイミングにおいて、VLC復号部 1 5 6 および量子化係数合成部 1 5 7 に供給する。

【 0 2 8 9 】

符号分割部 1 5 1 は、ライン判定部 1 5 2、VLC復号部 1 5 4、VLC復号部 1 5 5、およびVLC復号部 1 5 6 のそれぞれから供給される情報に基づいて、入力された、符号化された画像としての符号を分割し、分割された所定の長さの符号をライン判定部 1 5 2、VLC復号部 1 5 4、VLC復号部 1 5 5、またはVLC復号部 1 5 6 に供給する。

【 0 2 9 0 】

すなわち、符号分割部 1 5 1 は、入力された符号を、符号化された 1 つのラインの量子化係数が全て 0 であるか否かを示す符号、符号化された W 個の量子化係数の最大有効桁数を示す符号、符号化された W 個の量子化係数の絶対値を示す符号、および符号化された量子化係数のサインを示す符号に分割し、それぞれを、ライン判定部 1 5 2、VLC復号部 1 5 4、VLC復号部 1 5 5、およびVLC復号部 1 5 6 の内、各データに対応する処理部に供給する。

【 0 2 9 1 】

より具体的には、符号分割部 1 5 1 は、符号化された 1 つのラインの量子化係数が全て 0 であるか否かを示す符号を、ライン判定部 1 5 2 に供給する。また、符号分割部 1 5 1 は、符号化された W 個の量子化係数の最大有効桁数を示す符号を、VLC復号部 1 5 4 に供

10

20

30

40

50

給する。また、符号分割部 1 5 1 は、符号化された W 個の量子化係数の絶対値を示す符号を、VLC復号部 1 5 5 に供給する。さらに、符号分割部 1 5 1 は、符号化された量子化係数のサインを示す符号を、VLC復号部 1 5 6 に供給する。

【 0 2 9 2 】

ライン判定部 1 5 2 は、符号分割部 1 5 1 から供給された符号に基づいて、符号化されたサブバンドの 1 つのラインの量子化係数が全て 0 であるか否かを判定し、その判定の結果を示す情報を、符号分割部 1 5 1、発生部 1 5 3、および VLC復号部 1 5 4 に供給する。

【 0 2 9 3 】

発生部 1 5 3 は、ライン判定部 1 5 2 からの判定の結果を示す情報に基づいて、1 ライン分の 0 である量子化係数を示す符号を発生して切替部 1 5 8 に供給する。

10

【 0 2 9 4 】

VLC復号部 1 5 4 は、符号分割部 1 5 1 から供給された、符号化された W 個の量子化係数の最大有効桁数を示す符号を復号して、符号化された W 個の量子化係数の最大有効桁数を求め、求めた最大有効桁数を示す情報を、符号分割部 1 5 1 に供給する。また、VLC復号部 1 5 4 は、求めた最大有効桁数を示す情報を、遅延部 1 6 1 を介して、VLC復号部 1 5 5 および量子化係数合成部 1 5 7 に供給する。

【 0 2 9 5 】

また、VLC復号部 1 5 4 は、符号分割部 1 5 1 から供給された、ゼロランを示す符号を復号し、ゼロランを構成する量子化係数群を生成する。VLC復号部 1 5 4 は、復号されて得られた値が 0 の量子化係数群を、符号分割部 1 5 1 に供給する。また、VLC復号部 1 5 4 は、生成した値が 0 の量子化係数群を、遅延部 1 6 1 を介して、VLC復号部 1 5 5 および量子化係数合成部 1 5 7 に供給する。

20

【 0 2 9 6 】

符号分割部 1 5 1 乃至 VLC復号部 1 5 4 は、符号化データの入カタイミングにおいて、処理を行う。

【 0 2 9 7 】

VLC復号部 1 5 5 は、遅延部 1 6 1 を介して VLC復号部 1 5 4 から供給された最大有効桁数を示す情報に基づいて、符号分割部 1 5 1 から供給された量子化係数の絶対値を示す符号を復号する。VLC復号部 1 5 5 は、これにより得られた W 個の量子化係数の有効桁 (のデータ) を、遅延部 1 6 2 を介して、VLC復号部 1 5 6 および量子化係数合成部 1 5 7 に供給する。また、VLC復号部 1 5 5 は、量子化係数の絶対値を示す符号の復号の結果を示す情報を符号分割部 1 5 1 に供給する。

30

【 0 2 9 8 】

つまり、VLC復号部 1 5 5 は、符号化データの入カタイミングから 1 組分遅延して、処理を行う。

【 0 2 9 9 】

VLC復号部 1 5 6 は、遅延部 1 6 2 を介して VLC復号部 1 5 5 から供給された量子化係数の有効桁に基づいて、符号分割部 1 5 1 から供給された量子化係数のサインを示す符号を復号し、これにより得られる量子化係数のサイン (のデータ) を量子化係数合成部 1 5 7 に供給する。また、VLC復号部 1 5 6 は、量子化係数のサインを示す符号の復号の結果を示す情報を符号分割部 1 5 1 に供給する。

40

【 0 3 0 0 】

量子化係数合成部 1 5 7 は、遅延部 1 6 1 を介して VLC復号部 1 5 4 から供給された最大有効桁数を示す情報に基づいて、遅延部 1 6 2 を介して VLC復号部 1 5 5 から供給された量子化係数の絶対値と、VLC復号部 1 5 6 から供給された量子化係数のサインとを合成し、これにより得られた W 個の量子化係数を切替部 1 5 8 に供給する。

【 0 3 0 1 】

つまり、VLC復号部 1 5 6 および量子化係数合成部 1 5 7 は、符号化データの入カタイミングから 2 組分遅延して、処理を行う。

50

【0302】

切替部158は、発生部153または量子化係数合成部157からの量子化係数を出力する。

【0303】

[符号分割部]

図25は、符号分割部151の主な構成例を示すブロック図である。図25に示されるように、符号分割部151は、制御部171およびメモリ172を有する。

【0304】

制御部171は、図24に示されるライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれから供給される情報に基づいて、メモリ172に一時的に記憶されている符号のうち、所定の長さの符号を読み出して、ライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給する。

10

【0305】

また、符号分割部151は、図25に示した構成例の他、例えば、図26に示されるように構成されてもよい。

【0306】

図26に示される符号分割部151は、制御部191、スイッチ192、並びに、ノード193-1乃至ノード193-4を有する。

【0307】

制御部191は、符号化された画像としての符号が符号分割部151に入力されると、図24に示したライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれから供給される情報に基づいて、スイッチ192を制御し、入力された符号のうち、所定の長さの符号をライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給させる。

20

【0308】

すなわち、ノード193-1乃至ノード193-4のそれぞれは、ライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれに接続されており、制御部191は、符号の供給先として、ノード193-1乃至ノード193-4のいずれかを選択し、スイッチ192と、選択されたノードとの接続を制御する。

【0309】

スイッチ192が制御部191の制御に基づいて選択されたノードを入力と接続するので、符号分割部151に入力された符号は、スイッチ192、およびスイッチ192と接続されたノードを介して、符号の供給先として選択されたライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給される。

30

【0310】

[VLC復号部]

図27は、VLC復号部154の主な構成例を示すブロック図である。

【0311】

図27に示されるように、VLC復号部154は、最大有効桁数変化復号部201、最大有効桁数変化量復号部202、および0連長復号部203を有する。

40

【0312】

最大有効桁数変化復号部201は、符号分割部151から供給される、最大有効桁数を示す符号に含まれる、最大有効桁数の変化を示す符号を復号する。

【0313】

最大有効桁数変化量復号部202は、符号分割部151から供給される、最大有効桁数を示す符号に含まれる、最大有効桁数の変化量を示す符号を復号する。

【0314】

VLC復号部154は、それらの復号により得られた情報を、VLC復号部155や量子化係数合成部157に供給する。

【0315】

50

また、0連長復号部203は、符号分割部151から供給される、ゼロランを示す符号を復号し、ゼロランを構成する量子化係数群を生成する。VLC復号部154は、生成された量子化係数群を、VLC復号部155および量子化係数合成部157に供給する。

【0316】

[パイプライン処理に対応したゼロラン復号処理]

本技術においては、最大有効桁数Bの復号に2つの状態、ノーマルモード(normal mode)とゼロランモード(zero run mode)を導入する。各ラインの復号開始時に、状態はノーマルモード(normal mode)に初期化される。

【0317】

ノーマルモード(normal mode)では、画像復号装置111は、図13の例の場合と同様に、次のBを1つだけ復号する。そして、Bが0になることが明らかになったら、ゼロランモード(zero run mode)に移る(3)。

10

【0318】

ゼロランモード(zero run mode)では、ゼロラン符号により0連長を復号し、ライン最後まで0ではない場合、画像復号装置111は、続けてmore*, enoughの符号により、次の0でないBの値を復号する(4)。そして0連長だけ進んだあとでノーマルモード(normal mode)に移る。

【0319】

画像復号装置111が(3)で復号するのは、(1)で符号化した符号である。そして、画像復号装置111は、(3)の復号を終えた後の符号を使って(3)の復号と同じタイミングでS(3)を復号し、次のタイミングで(2)で符号化した符号を(4)で復号する。したがって、画像復号装置111は、画像符号化装置11が符号化した符号化データを復号することができる。

20

【0320】

[復号処理の流れ]

次に、図28のフローチャートを参照して、画像復号装置111による復号処理について説明する。この復号処理は、エントロピ復号部121に符号化された画像としての符号が入力されると開始される。

【0321】

ステップS131において、エントロピ復号部121は、エントロピ復号処理を行い、入力された画像としての符号をエントロピ復号し、これにより得られた量子化係数を逆量子化部122に供給する。

30

【0322】

なお、エントロピ復号処理の詳細は後述するが、このエントロピ復号処理において、エントロピ復号部121は、符号化されたサブバンドのライン上の連続する位置の量子化係数をW個ずつ復号し、復号された量子化係数を逆量子化部122に供給する。また、このエントロピ復号処理において、エントロピ復号部121は、値が0の量子化係数よりなるゼロランを示す符号を復号することにより、ゼロランを構成する値が0の量子化係数群をまとめて復号する。

【0323】

その際、エントロピ復号部121は、エントロピ復号処理の各処理要素を、図15を参照して説明したようなタイミングで、互いに並列に実行する。これによりエントロピ復号部121は、エントロピ復号処理のスループットを向上させることができる。

40

【0324】

ステップS132において、逆量子化部122は、エントロピ復号部121から供給された量子化係数を逆量子化し、逆量子化により得られた各サブバンドのウェーブレット係数をウェーブレット逆変換部123に供給する。

【0325】

ステップS133において、ウェーブレット逆変換部123は、逆量子化部122から供給された、各サブバンドのウェーブレット係数にウェーブレット逆変換を施し、その結

50

果得られた画像を出力して復号処理は終了する。

【0326】

このようにして、画像復号装置111は、符号化された画像を復号して出力する。

【0327】

[エントロピ復号処理の流れ]

次に、図29のフローチャートを参照して、図28のステップS131の処理に対応するエントロピ復号処理について説明する。

【0328】

ステップS161において、ライン判定部152は、これから復号するサブバンドのラインを示す変数 y を $y=0$ として、これを記憶する。

10

【0329】

ステップS162において、VLC復号部154は、ライン判定部152が記憶している変数 y により示されるライン y よりも1つ前のライン($y-1$)上の最初に入力される W 個の量子化係数の最大有効桁数を示す変数 B_{init} を $B_{init}=0$ として、これを記憶する。

【0330】

例えば、ライン($y-1$)が、図2に示したライン $L1$ である場合、ライン($y-1$)上の最初に入力される W 個の量子化係数の最大有効桁数を示す変数 B_{init} の値は、ライン $L1$ の図中、左端の位置から W 個の量子化係数の最大有効桁数となる。また、ライン判定部152が記憶している変数 y が $y=0$ である場合、ライン($y-1$)は存在しないので、変数 B_{init} の値は、 $B_{init}=0$ とされる。

20

【0331】

ステップS163において、符号分割部151は、入力された符号のうち、最初の1ビットの符号を、これから復号するラインの量子化係数が全て0であるか否かを示す符号として、ライン判定部152に供給する。

【0332】

また、そのステップS163において、ライン判定部152は、符号分割部151から読み込んだ(供給された)1ビットの符号が0であるか否かを判定し、その判定の結果を示す情報を生成して発生部153、VLC復号部154、および符号分割部151に供給する。符号が0であると判定した場合、ライン y の量子化係数が全て0であるので、ライン判定部152は、処理をステップS164に進める。

30

【0333】

ステップS164において、発生部153は、ライン判定部152からの判定の結果を示す情報に基づいて、ライン y の量子化係数を全て0とする。そして、発生部153は、ライン y の量子化係数を示す符号を発生して切替部158に供給する。

【0334】

例えば、1つの量子化係数が4桁で表わされ、1つのラインの量子化係数が5個である場合、発生部153は、ライン y の量子化係数を示す符号として、 $20 (= 4 \times 5)$ 個の0を発生して切替部158に供給する。切替部158は、発生部153から供給された連続する20個の0を、1つのラインの量子化係数を示す符号として、逆量子化部122に出力する。

40

【0335】

ステップS165において、VLC復号部154は、ライン判定部152からの判定の結果を示す情報に基づいて、記憶している変数 B_{init} の値を $B_{init}=0$ とし、変数 B_{init} を更新する。

【0336】

ステップS166において、ライン判定部152は、復号しているサブバンドのラインのうち、未処理のラインがあるか否かを判定する。すなわち、ライン判定部152は、復号しているサブバンドの全てのライン上の位置の量子化係数を復号したか否かを判定する。

【0337】

50

ステップS 1 6 6において、未処理のラインがあると判定した場合、ライン判定部 1 5 2は、自分自身が記憶している変数 y により示されるライン y の次のライン($y + 1$)上の各位置の量子化係数を復号するので、処理をステップS 1 6 7に進める。

【0338】

ステップS 1 6 7において、ライン判定部 1 5 2は、記憶しているラインを示す変数 y をインクリメントして $y = y + 1$ とし、処理をステップS 1 6 3に戻し、それ以降の処理を実行させる。

【0339】

これに対して、ステップS 1 6 6において、未処理のラインが無いと判定した場合、サブバンドを構成する全てのラインについて量子化係数が復号されたので、ライン判定部 1 5 2は、エントロピ復号処理を終了し、処理を図28のステップS 1 3 1に戻し、ステップS 1 3 2以降の処理を実行させる。

10

【0340】

また、図29のステップS 1 6 3において、符号が0でないと判定した場合、ライン判定部 1 5 2は、処理をステップS 1 6 8に進める。ステップS 1 6 8において、エントロピ復号部 1 2 1は、 W 個組復号処理を行う。

【0341】

なお、 W 個組復号処理の詳細は後述するが、この W 個組復号処理において、エントロピ復号部 1 2 1は、ライン判定部 1 5 2が記憶している変数 y により示されるライン y 上の連続する W 個の位置の量子化係数を復号する。

20

【0342】

また、この W 個組復号処理において、エントロピ復号部 1 2 1は、値が0の量子化係数よりなるゼロランを示す符号を復号することにより、ゼロランを構成する値が0の量子化係数群をまとめて復号する。

【0343】

W 個組復号処理が終了すると、エントロピ復号部 1 2 1は、処理をステップS 1 6 6に戻し、それ以降の処理を実行させる。

【0344】

このようにして、エントロピ復号部 1 2 1は、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に復号する。

30

【0345】

このように、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に復号することによって、符号化された量子化係数を、入力された順番に処理することができ、量子化係数の復号により生じる遅延をより少なくすることができる。

【0346】

[W 個組復号処理の流れ]

次に、図30のフローチャートを参照して、図29のステップS 1 6 8において実行される W 個組復号処理の流れの例を説明する。

【0347】

ステップS 1 8 1において、エントロピ復号部 1 2 1は、変数 i 、 B 、 \max 等、各種変数を初期化する。例えば、エントロピ復号部 1 2 1は、変数 $i = 0$ とし、変数 $B = 0$ とし、変数 $\max =$ 係数 W 個組の数とする。また、エントロピ復号部 1 2 1は、例えば、すべての係数の初期値 = 0とする。

40

【0348】

ステップS 1 8 2において、VLC復号部 1 5 6は、($i - 1$)番目の係数 W 個組が存在する場合、符号分割部 1 5 1から供給された今の組のサインを示す符号を、絶対値が0でない各量子化係数のサインを示す符号を並べたものであるとする。VLC復号部 1 5 6は、求めた各量子化係数のサインを、符号分割部 1 5 1と量子化係数合成部 1 5 7に供給する。

【0349】

50

量子化係数合成部 157 は、($i - 1$) 番目の係数 W 個組について、量子化係数の絶対値とサインとを合成し、これにより得られた W 個の量子化係数を、切替部 158 を介して出力する。

【 0350 】

なお、($i - 1$) 番目の係数 W 個組が存在しない場合、この処理は省略 (スキップされる) 。

【 0351 】

ステップ S 183 において、VLC復号部 155 は、 i 番目の係数 W 個組が存在する場合、その各係数について、前回のステップ S 184 の処理により求められた最大有効桁数 B ビット分の符号列を読み込み、それを、各量子化係数の最大有効桁数分の絶対値を並べたものとする。VLC復号部 155 は、求めた各量子化係数の絶対値を、符号分割部 151 に供給するとともに、遅延部 162 を介して、VLC復号部 156 および量子化係数合成部 157 に供給する。

10

【 0352 】

なお、 i 番目の係数 W 個組が存在しない場合、この処理は省略 (スキップされる) 。

【 0353 】

ステップ S 184 において、VLC復号部 154 は、最大有効桁数処理を行い、最大有効桁数を示す符号を復号して最大有効桁数 B を求める。VLC復号部 154 は、求めた最大有効桁数 B を、符号分割部 151 に供給するとともに、遅延部 161 を介して、VLC復号部 155 および量子化係数合成部 157 に供給する。

20

【 0354 】

ステップ S 184 の処理が終了すると、VLC復号部 154 は、処理をステップ S 185 に進める。

【 0355 】

ステップ S 185 において、エントロピ復号部 121 は、変数 i ($\max + 1$) であるか否かを判定する。すなわち、エントロピ復号部 121 は、処理対象のラインの全ての量子化係数の組を処理したか否かを判定する。

【 0356 】

変数 i ($\max + 1$) であり、処理対象のライン上に未処理の組が存在すると判定された場合、エントロピ復号部 121 は、処理をステップ S 186 に進める。

30

【 0357 】

ステップ S 186 において、エントロピ復号部 121 は、変数 i をインクリメント (変数 $i = i + 1$) し、処理をステップ S 182 に戻す。

【 0358 】

つまり、ステップ S 182 の処理は、図 24 の遅延部 161 および遅延部 162 により、ステップ S 184 の処理に比べて、2 周期 (2cycle) 分遅延されている。したがって、ステップ S 182 においては、ステップ S 184 の 2 組前の処理対象の組が処理対象とされる。

【 0359 】

また、ステップ S 183 の処理は、図 24 の遅延部 161 により、ステップ S 184 の処理に比べて、1 周期 (1cycle) 分遅延されている。したがって、ステップ S 183 においては、ステップ S 184 の 1 組前の処理対象の組が処理対象とされる。

40

【 0360 】

変数 i は、0 以上の整数である。したがって、例えば、 $i = 0$ の場合、ステップ S 184 の処理のみが行われる。また、例えば、 $i = 1$ の場合、ステップ S 183 およびステップ S 184 の処理が行われる。2 $i < \max$ の場合、ステップ S 182 乃至ステップ S 184 の処理が行われる。 $i = \max$ の場合、ステップ S 182 およびステップ S 183 の処理が行われる。 $i = \max + 1$ の場合、ステップ S 182 の処理のみが行われる。

【 0361 】

そして、ステップ S 185 において、 $i > (\max + 1)$ であると判定した場合、エント

50

ロピ復号部 1 2 1 は、W 個組復号処理を終了し、処理を図 2 9 に戻す。

【 0 3 6 2 】

[最大有効桁数処理の流れ]

次に、図 3 1 および図 3 2 のフローチャートを参照して、図 3 0 のステップ S 1 8 4 において実行される最大有効桁数処理の流れの例を説明する。

【 0 3 6 3 】

最大有効桁数処理が開始されると、ステップ S 2 0 1 において、VLC 復号部 1 5 4 は、(i + 1) 番目の係数 W 個組が存在するか否かを判定する。存在しないと判定した場合、VLC 復号部 1 5 4 は、最大有効桁数処理を終了し、処理を図 3 0 に戻す。例えば、i max の場合、(i + 1) 番目の係数 W 個組は、存在しない。このような場合、最大有効桁数処理は終了される。

10

【 0 3 6 4 】

また、図 3 1 のステップ S 2 0 1 において、(i + 1) 番目の係数 W 個組が存在すると判定した場合、VLC 復号部 1 5 4 は、処理をステップ S 2 0 2 に進める。

【 0 3 6 5 】

ステップ S 2 0 2 において、0 連長復号部 2 0 3 は、変数 count = 0 であるか否かを判定する。ゼロラン中でなく、count = 0 であると判定した場合、0 連長復号部 2 0 3 は、処理をステップ S 2 0 3 に進める。

【 0 3 6 6 】

ステップ S 2 0 3 において、VLC 復号部 1 5 4 は、モードが 0 乃至 2 のいずれであるかを判定する。モードが 0 (mode = 0) であると判定した場合、VLC 復号部 1 5 4 は、処理をステップ S 2 0 4 に進める。

20

【 0 3 6 7 】

ステップ S 2 0 4 において、最大有効桁数変化復号部 2 0 1 は、符号分割部 1 5 1 から供給される符号を 1 ビット読み込み、その符号が 0 であるか否かを判定する。読み込んだ符号が 0 である場合、最大有効桁数 B は、前の組と同じである。したがって、最大有効桁数変化復号部 2 0 1 は、ステップ S 2 0 5 乃至ステップ S 2 0 9 の各処理を省略し、処理をステップ S 2 1 1 に進める。

【 0 3 6 8 】

また、ステップ S 2 0 4 において、読み込んだ符号が 1 であると判定した場合、最大有効桁数 B の値が前の組から変化しているため、最大有効桁数変化復号部 2 0 1 は、処理をステップ S 2 0 5 に進める。

30

【 0 3 6 9 】

ステップ S 2 0 5 において、最大有効桁数変化復号部 2 0 1 は、符号分割部 1 5 1 から供給される符号を 1 ビット読み込み、その符号が 0 であるか否かを判定する。読み込んだ符号が 0 であると判定した場合、最大有効桁数 B の値は、前の組より増加している。この場合、最大有効桁数変化復号部 2 0 1 は、ステップ S 2 0 6 に処理を進める。

【 0 3 7 0 】

ステップ S 2 0 6 において、最大有効桁数変化量復号部 2 0 2 は、符号 1 を読み込むまで、符号分割部 1 5 1 から供給される符号を 1 ビットずつ読み込む。最大有効桁数変化量復号部 2 0 2 は、符号 1 を読み込んだ時点で、この読み込みを終了し、それまでに読み込んだ符号 0 の数を n とする。

40

【 0 3 7 1 】

ステップ S 2 0 7 において、最大有効桁数変化量復号部 2 0 2 は、その変数 n を用いて、最大有効桁数 B の値を更新する。すなわち、最大有効桁数変化量復号部 2 0 2 は、処理対象である今の組の最大有効桁数 B を、以下の式 (1) のように決定する。

【 0 3 7 2 】

$$B = B + (n + 1) \cdot \dots \cdot (1)$$

【 0 3 7 3 】

今の組の最大有効桁数 B を決定すると、最大有効桁数変化量復号部 2 0 2 は、処理をス

50

ステップ S 2 1 1 に進める。

【 0 3 7 4 】

また、ステップ S 2 0 5 において、読み込んだ符号が 0 でないと判定した場合、最大有効桁数 B の値は、前の組より減少している。この場合、最大有効桁数変化復号部 2 0 1 は、ステップ S 2 0 8 に処理を進める。

【 0 3 7 5 】

ステップ S 2 0 8 において、最大有効桁数変化量復号部 2 0 2 は、符号 1 を読み込むまで、若しくは、最大 (B - 1) ビット符号を読み込むまで、符号分割部 1 5 1 から供給される符号を 1 ビットずつ読み込む。最大有効桁数変化量復号部 2 0 2 は、符号 1 を読み込んだ時点、若しくは、(B - 1) ビット符号を読み込んだ時点で、この読み込みを終了し、それまでに読み込んだ符号 0 の数を n とする。

10

【 0 3 7 6 】

ステップ S 2 0 9 において、最大有効桁数変化量復号部 2 0 2 は、その変数 n を用いて、最大有効桁数 B の値を更新する。すなわち、最大有効桁数変化量復号部 2 0 2 は、処理対象である今の組の最大有効桁数 B を、以下の式 (2) のように決定する。

【 0 3 7 7 】

$$B = B - (n + 1) \quad \cdots (2)$$

【 0 3 7 8 】

今の組の最大有効桁数 B を決定すると、最大有効桁数変化量復号部 2 0 2 は、処理をステップ S 2 1 1 に進める。

20

【 0 3 7 9 】

また、ステップ S 2 0 3 において、モードが 2 (mode = 2) であると判定した場合、VLC 復号部 1 5 4 は、処理をステップ S 2 1 0 に進める。

【 0 3 8 0 】

ステップ S 2 1 0 において、VLC 復号部 1 5 4 は、モードを 0 (mode = 0) に戻し、処理をステップ S 2 0 6 に戻し、それ以降の処理を実行させる。

【 0 3 8 1 】

ステップ S 2 1 1 において、VLC 復号部 1 5 4 は、最大有効桁数 B が 0 である場合、すなわち、ゼロランになった場合、モードを 1 (mode = 1) にする。

【 0 3 8 2 】

ステップ S 2 1 1 の処理を終了すると、VLC 復号部 1 5 4 は、最大有効桁数処理を終了し、処理を図 3 0 に戻す。

30

【 0 3 8 3 】

また、図 3 1 のステップ S 2 0 2 において、変数 count = 0 でないと判定した場合、ゼロラン中であるので、0 連長復号部 2 0 3 は、処理をステップ S 2 1 2 に進める。

【 0 3 8 4 】

ステップ S 2 1 2 において、0 連長復号部 2 0 3 は、以下の式 (3) のように、変数 count をデクリメント (1 減算) する。

【 0 3 8 5 】

$$\text{count} = \text{count} - 1 \quad \cdots (3)$$

40

【 0 3 8 6 】

ステップ S 2 1 3 において、VLC 復号部 1 5 4 は、この変数 count が 0 であれば、モードを 2 (mode = 2) に設定する。

【 0 3 8 7 】

ステップ S 2 1 3 の処理が終了すると、VLC 復号部 1 5 4 は、最大有効桁数処理を終了し、処理を図 3 0 に戻す。

【 0 3 8 8 】

また、図 3 1 のステップ S 2 0 3 において、モードが 1 (mode = 1) であると判定した場合、VLC 復号部 1 5 4 は、処理を図 3 2 に進める。

【 0 3 8 9 】

50

図32のステップS231において、VLC復号部154は、モードを0 (mode = 0) に設定する。また、ステップS232において、0連長復号部203は、変数max - i - 1の桁数をDとする。

【0390】

ステップS233において、0連長復号部203は、符号の読み込み(符号分割部151からの符号の取得)を開始し、符号1を取得するか、最大Dビット読み込むまで、その符号の読み込み(1回目の読み込み)を行う。

【0391】

この1回目の読み込みにより、図5のテーブルの一番右の列に示される符号列の、カンマ(,)より左の部分が読み込まれるか、または、図7若しくは図9のテーブルの左から3番目の列の符号列が読み込まれる。

10

【0392】

そして、0連長復号部203は、読み込んだ符号0の数をnとする。

【0393】

ステップS234において、0連長復号部203は、D = nであるか否かを判定する。D = nであると判定された場合、すなわち、ゼロランがラインの最後まで達していないと判定された場合、0連長復号部203は、処理をステップS235に進める。

【0394】

つまり、この場合、1回目の読み込みにより、図5のテーブルの一番右の列に示される符号列の、カンマ(,)より左の部分が読み込まれていることになる。この1回目の読み込みにより、そこからn個目の符号までが、カンマ(,)の右の部分に相当するということが明らかとなっている。

20

【0395】

そこで、0連長復号部203は、ステップS235において、符号をnビット読み込む(2回目の読み込み)。その読み込んだnビットの先頭に符号1をつけた符号列を変数countに代入する。つまり、図5のテーブルの左から2列目の2進表現が変数countに代入される。

【0396】

ステップS235の処理が終了すると、0連長復号部203は、処理を図31のステップS212に戻し、それ以降の処理を実行させる。

30

【0397】

また、図32のステップS234において、D = nであると判定された場合、すなわち、ゼロランがラインの最後まで達すると判定された場合、0連長復号部203は、処理をステップS236に進める。

【0398】

つまり、この場合、1回目の読み込みにより、図7若しくは図9のテーブルの左から3番目の列の符号列が読み込まれていることになる。

【0399】

そこで、0連長復号部203は、ステップS236において、変数countの値を(max - 1)とし、処理を図31のステップS212に戻し、それ以降の処理を実行させる。

40

【0400】

このように、符号化されたサブバンドの量子化係数を、予め定められた所定の数ずつまとめて復号することで、例えば、JPEG2000方式により画像を復号する場合とは異なり、複数のコーディングパスに基づいて、ビットプレーンごとに複数の処理を行う必要がなく、より高速に画像の復号を行うことができる。これにより、高解像度の画像をリアルタイムで復号するための復号装置を安価で実現することができる。

【0401】

また、以上のように各処理を実行することにより、画像復号装置111は、ゼロランがまとめて符号化された符号を正しく復号し、値が0の量子化係数を復元することができる。したがって、画像復号装置111は、ゼロランの符号量の低減を実現することができ、

50

0が多い係数データの符号化の符号化効率の向上を実現することができる。

【0402】

さらに、上述したように、復号処理の各処理要素の実行を制御し、1つの組に対する復号処理の各処理要素を、互いに異なるタイミングで実行させるようにすることにより、画像復号装置111は、復号処理のスループットを向上させることができる。その結果、画像復号装置111は、より高速に復号処理を行うことができる。

【0403】

<3.第3の実施の形態>

[パイプライン処理に適応したゼロラン符号化処理の他の例]

なお、ゼロランの符号化は、上述した例に限らず、以下のように行っても良い。

10

【0404】

例えば、上述した(2)において、最大有効桁数Bが0に変化した直後の係数でBが0でない場合、符号1に接続してその0でなかったBの値をmore*, enoughの符号で送る。

【0405】

これに対して、Bが0に変化した直後の係数でBが0であった場合、符号0を送る。0連長が確定した時点で、ゼロラン符号により0連長を符号化したものから符号先頭の1ビットの0を削除して送り、ライン最後まで0ではない場合、続けてmore*, enoughの符号により、次の0でないBの値を符号化する。

【0406】

20

このようにしても、第1の実施の形態と同様に符号化を行うことができる。

【0407】

したがって、この場合も、画像符号化装置11の構成も、各処理の流れも、第1の実施の形態において説明した場合と、基本的に同様である。

【0408】

ただし、最大有効桁数処理において、図21のフローチャートの各処理は、第1の実施の形態の場合と同様に実行されるが、図22のフローチャートの部分の各処理は、図33のフローチャートに示されるように実行される。

【0409】

つまり、図21のステップS73において、変数count = 0でない判定された場合、処理は、図33のステップS301に進められる。そして、そのステップS301において、0連長符号化部83は、さらに、変数count = 1であるか否かを判定する。変数count = 1である場合、すなわち、Bが0に変化した直後の組の場合、0連長符号化部83は、処理をステップS302に進める。

30

【0410】

ステップS302において、0連長符号化部83は、Bnew = 0であるか否かを判定する。Bnew = 0であると判定した場合、0連長符号化部83は、処理をステップS303に進め、符号0を出力する。そして、ステップS304において、0連長符号化部83は、変数countをインクリメントし(1加算し)、以下の式(4)のように更新する。

【0411】

40

count = count + 1 . . . (4)

【0412】

ステップS304の処理が終了すると、0連長符号化部83は、処理を図21のステップS82に戻す。

【0413】

また、図33のステップS302において、Bnew = 0でない判定した場合、0連長符号化部83は、処理をステップS305に進め、符号1を出力する。

【0414】

ステップS306において、0連長符号化部83は、(Bnew - 1)個の0に続いて、符号1を出力する。

50

【 0 4 1 5 】

ステップ S 3 0 7 において、0 連長符号化部 8 3 は、変数 count の値を 0 (count = 0) とする。ステップ S 3 0 7 の処理が終了すると、0 連長符号化部 8 3 は、処理を図 2 1 のステップ S 8 2 に戻す。

【 0 4 1 6 】

また、ステップ S 3 0 1 において、変数 count = 1 でないと判定した場合、すなわち、B が 0 に変化した直後の組でない場合、0 連長符号化部 8 3 は、処理をステップ S 3 0 8 に進める。

【 0 4 1 7 】

図 3 3 のステップ S 3 0 8 において、0 連長符号化部 8 3 は、Bnew = 0 であるか否かを判定する。Bnew = 0 であり、今の組でもゼロランが終了しないと判定された場合、0 連長符号化部 8 3 は、処理をステップ S 3 0 9 に進める。

10

【 0 4 1 8 】

ステップ S 3 0 9 において、0 連長符号化部 8 3 は、ゼロランの組数のカウント値である変数 count をインクリメント (変数 count = count + 1) し、処理を図 2 1 のステップ S 8 2 に戻し、それ以降の処理を実行させる。

【 0 4 1 9 】

また、図 3 3 のステップ S 3 0 8 において、Bnew = 0 でなく、今の組でゼロランが終了すると判定された場合、0 連長符号化部 8 3 は、処理をステップ S 3 1 0 に進める。

【 0 4 2 0 】

20

ステップ S 3 1 0 において、0 連長符号化部 8 3 は、例えば図 5 のテーブルの一番右の列に示されるような、(変数 count の桁数 - 2) 個の符号 0 に、変数 count の 2 進表現を続けたものを、ゼロランを示す符号として符号連結部 6 9 に出力する。

【 0 4 2 1 】

ステップ S 3 1 1 において、最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B の増分を示す (Bnew - 1) 個の符号 0 を符号連結部 6 9 に出力する。なお、ここでは、変動の終端 (enough) を示す符号 1 の出力は省略 (スキップ) されている。この変動の終端 (enough) を示す符号 1 の出力を省略しない場合、最大有効桁数変化量符号化部 8 2 は、最大有効桁数 B の増分を示す (Bnew - 1) 個の符号 0 に続けて、その変動の終端 (enough) を示す符号 1 を符号連結部 6 9 に出力すればよい。

30

【 0 4 2 2 】

ステップ S 3 1 2 において、0 連長符号化部 8 3 は、変数 count の値を 0 とする。ステップ S 3 1 2 の処理が終了すると、0 連長符号化部 8 3 は、処理を図 2 1 のステップ S 8 2 に戻し、それ以降の処理を実行させる。

【 0 4 2 3 】

このようにして、エントロピ符号化部 2 3 は、第 1 の実施の形態の場合と同様に符号化を行うことができる。

【 0 4 2 4 】

[パイプライン処理に適応したゼロラン復号処理の他の例]

この符号化に対応する復号処理について説明する。ゼロランモード (zero run mode) の最初のタイミングにおいて、画像復号装置 1 1 1 は、0 若しくは 1 , more* , enough のどちらかを復号する。

40

【 0 4 2 5 】

1 , more* , enough を復号した場合、0 連長が 1 であり、more* , enough によって示される値が新しい B である。0 連長が 1 であるからこのタイミングで B の値が必要となるが、きちんと復号出来ている。

【 0 4 2 6 】

これに対して、0 を復号した場合、画像復号装置 1 1 1 は、ゼロランモード (zero run mode) の次のタイミングで既に 0 を 1 ビット使っていることを考慮に入れてゼロラン符号により 0 連長を復号する。つまり、画像復号装置 1 1 1 は、ライン最後まで 0 ではない

50

場合、続けてmore* , enoughの符号により、次の0でないBの値を復号する。そして画像復号装置111は、0連長だけ進んだあとでノーマルモード(normal mode)に移る。0連長が2以上なので、このタイミングで0連長を復号しても問題ない。

【0427】

この場合も、画像復号装置111の構成も、各処理の流れも、第2の実施の形態において説明した場合と、基本的に同様である。

【0428】

ただし、最大有効桁数処理は、図34および図35のフローチャートのよう実行される。

【0429】

図34に示されるステップS351乃至ステップS363の各処理は、基本的に図31のステップS201乃至ステップS213の各処理と同様に実行される。

【0430】

図34のステップS353において、モードが1(mode = 1)であると判定した場合、VLC復号部154は、処理を図35に進める。

【0431】

図35のステップS381において、最大有効桁数変化復号部201は、符号分割部151から供給される符号を1ビット読み込み、その符号が0であるか否かを判定する。

【0432】

読み込んだ符号が0であると判定した場合、最大有効桁数変化復号部201は、処理をステップS382に進める。ステップS382乃至ステップS387の各処理は、図32のステップS231乃至ステップS236の各処理と同様に実行される。

【0433】

また、ステップS381において、読み込んだ符号が1であると判定した場合、最大有効桁数変化復号部201は、処理をステップS388に進める。

【0434】

ステップS388において、VLC復号部154は、モードを2(mode = 2)に設定する。

【0435】

ステップS388の処理が終了すると、VLC復号部154は、処理を図34のステップS353に戻す。

【0436】

このようにして、エントロピ復号部121は、第2の実施の形態の場合と同様に復号を行うことができる。

【0437】

なお、復号処理の各処理要素の符号化データを、図13の例のように並べる場合、図30のフローチャートにおいて、ステップS182乃至ステップS184の各処理の実行順を、ステップS184、ステップS183、ステップS182の順に入れ替えればよい。

【0438】

なお、以上においては、図12や図14の例のように、絶対値の符号化処理または復号処理Aの実行タイミングを、最大有効桁数の符号化処理または復号処理Bより1組分遅延させ、サインの符号化処理または復号処理Sの実行タイミングを、最大有効桁数の符号化処理または復号処理Bより2組分遅延させるように説明した。しかしながら、各処理要素の実行タイミングは、互いに異なっていればよく、このパターンに限らない。

【0439】

例えば、絶対値の符号化処理または復号処理Aの実行タイミングを、最大有効桁数の符号化処理または復号処理Bより2組分以上遅延させてもよい。その場合、サインの符号化処理または復号処理Sの実行タイミングは、絶対値の符号化処理または復号処理Aより0組分以上遅延させればよい。

【0440】

10

20

30

40

50

また、例えば、サインの符号化処理または復号処理 S の実行タイミングは、絶対値の符号化処理または復号処理 A より 2 組分以上遅延させてもよい。

【 0 4 4 1 】

さらに、以上においては、時刻 t_i を、処理要素の実行タイミングとして説明したが、符号化処理や復号処理の各処理要素の実行タイミングは、時刻 t_i に限らない。符号化処理や復号処理の各処理要素は、図 1 2 や図 1 4 のタイミングチャートで示した依存関係による遅延が生じない限り、任意のタイミングで実行することができる。

【 0 4 4 2 】

例えば、時刻 t_i から時刻 $t(i+1)$ までの時間を 1 周期とし、その周期の中で、処理を終了することができるタイミングであれば、任意のタイミングで処理要素が実行されるようにしてもよい。

【 0 4 4 3 】

< 4 . 第 4 の実施の形態 >

[パイプライン処理の他の例]

また、次の周期に跨って、1 処理要素の処理が行われるようにしてもよい。つまり、1 処理要素の処理を、複数周期を使って実行するようにしてもよい。例えば、最大有効桁数の符号化処理または復号処理 B を、例えば、図 3 6 に示されるように、2 組分の時間を用いて行うようにしても良い。この場合、符号化データにおける各処理要素の符号の並びは、図 3 6 の点線矢印に示されるようになる。つまり、この場合も、符号の並びは、図 1 2 の場合と同様である。

【 0 4 4 4 】

その場合も、依存関係から、図 3 6 に示されるように、最大有効桁数の符号化処理または復号処理 B(i) の全ての処理は、絶対値の符号化処理または復号処理 A(i) を行う前に終了させる必要がある。例えば、最大有効桁数の符号化処理または復号処理 B の処理時間が、絶対値の符号化処理または復号処理 A やサインの符号化処理または復号処理 S の処理時間に比べて極端に長い場合（処理時間の偏りが大きい場合）、このように処理を行うことで、各サイクルに発生する待機時間を低減させることができ、スループットを向上させることができる。

【 0 4 4 5 】

この場合、エントロピ符号化部 2 3 は、図 3 7 に示される例のように構成すればよい。すなわち、この場合、エントロピ符号化部 2 3 は、図 1 6 の構成に加え、VLC符号化部 3 0 1 および遅延部 3 0 2 をさらに有する。

【 0 4 4 6 】

VLC符号化部 3 0 1 は、VLC符号化部 6 4 と合わせて、図 1 6 の場合のVLC符号化部 6 4 と同様の処理を行う。つまり、図 1 6 の場合、のVLC符号化部 6 4 が、図 2 1 および図 2 2 のフローチャートに示される最大有効桁数処理の各処理を実行するが、図 3 7 の例の場合、これらの処理の内、一部の処理をVLC符号化部 3 0 1 が行い、残りの処理をVLC符号化部 6 4 が行う。

【 0 4 4 7 】

遅延部 3 0 2 は、遅延部 7 1 乃至遅延部 7 5 と同様に、VLC符号化部 3 0 1 から出力される情報を、1 組分遅延させて、VLC符号化部 6 4 に供給する。

【 0 4 4 8 】

このようにすることにより、最大有効桁数処理が 2 組分のサイクルで実行されるようになる。なお、最大有効桁数処理のどの処理までをVLC符号化部 3 0 1 が行うようにし、どの処理からをVLC符号化部 6 4 が行うようにするかは任意である。

【 0 4 4 9 】

また、この場合、エントロピ復号部 1 2 1 は、図 3 8 に示される例のように構成すればよい。すなわち、この場合、エントロピ復号部 1 2 1 は、図 2 4 の構成に加え、VLC復号部 3 5 1 をさらに有する。

【 0 4 5 0 】

10

20

30

40

50

VLC復号部 3 5 1 は、VLC復号部 1 5 4 と合わせて、図 2 4 の場合のVLC復号部 1 5 4 と同様の処理を行う。つまり、図 2 4 の場合、のVLC復号部 1 5 4 が、図 3 1 および図 3 2 のフローチャートに示される最大有効桁数処理の各処理を実行するが、図 3 8 の例の場合、これらの処理の内、一部の処理をVLC復号部 1 5 4 が行い、残りの処理をVLC復号部 3 5 1 が行う。

【 0 4 5 1 】

VLC復号部 3 5 1 は、遅延部 1 6 1 を介して、VLC復号部 1 5 4 から、処理結果（図 3 1 および図 3 2 の最大有効桁数処理の途中結果）を取得し、残りの処理を実行する。

【 0 4 5 2 】

このようにすることにより、最大有効桁数処理が 2 組分のサイクルで実行されるようになる。なお、最大有効桁数処理のどの処理までをVLC復号部 1 5 4 が行うようにし、どの処理からをVLC復号部 3 5 1 が行うようにするかは任意である。

【 0 4 5 3 】

ところで、以上において説明した画像符号化装置 1 1 においては、量子化係数の絶対値を符号化（または復号）する場合に、予め定められたW個の量子化係数の絶対値を順番に符号化すると説明したが、汎用DSP（Digital Signal Processor）や、汎用CPUにおいて用いられているSIMD（Single Instruction Multiple Data）演算命令を利用して、W個の量子化係数を同時に（並列に）符号化（または復号）することにより、より高速に画像を符号化（または復号）することができる。

【 0 4 5 4 】

なお、エントロピ符号化部 2 3 が符号化するデータは、量子化係数以外であっても良い。例えば、画像符号化装置 1 1 において、量子化部 2 2 を省略し、エントロピ符号化部 2 3 が、ウェーブレット変換部 2 1 から出力されるウェーブレット係数を符号化するようにしてもよい。その場合、エントロピ復号部 1 2 1 は、入力された符号化データを復号し、ウェーブレット係数を出力する。したがって、この場合、画像復号装置 1 1 1 において、逆量子化部 1 2 2 を省略することができる。エントロピ復号部 1 2 1 は、入力された符号化データを復号し、画像データを出力する。したがって、この場合、画像復号装置 1 1 1 において、ウェーブレット逆変換部 1 2 3 も省略することができる。

【 0 4 5 5 】

また、さらにウェーブレット変換部 2 1 も省略し、エントロピ符号化部 2 3 が、画像データ（入力画像）を符号化するようにしてもよい。その場合も、エントロピ復号部 1 2 1 は、入力された符号化データを復号し、エントロピ符号化部 2 3 が符号化したデータに対応するデータを出力する。

【 0 4 5 6 】

< 5 . 第 5 の実施の形態 >

[画像符号化装置]

図 3 9 は、画像符号化装置の他の構成例を示すブロック図である。図 4 0 に示される画像符号化装置 4 0 1 は、入力される画像データを低遅延に符号化し、生成した符号化データを出力する画像処理装置である。

【 0 4 5 7 】

図 3 9 に示されるように、画像符号化装置 4 0 1 は、ウェーブレット変換部 4 1 0、途中計算用バッファ部 4 1 1、係数並び替え用バッファ部 4 1 2、係数並び替え部 4 1 3、レート制御部 4 1 4、およびエントロピ符号化部 4 1 5 を有する。

【 0 4 5 8 】

画像符号化装置 4 0 1 に入力された画像データは、途中計算用バッファ部 4 1 1 に格納される。ウェーブレット変換部 4 1 0 は、途中計算用バッファ部 4 1 1 に格納された画像データに対してウェーブレット変換を施す。すなわち、ウェーブレット変換部 4 1 0 は、途中計算用バッファ部 4 1 1 から画像データを読み出して、分析フィルタによりフィルタ処理を施して低域成分および高域成分の係数のデータを生成し、生成された係数データを途中計算用バッファ部 4 1 1 に格納する。

10

20

30

40

50

【0459】

ウェーブレット変換部410は、水平分析フィルタと垂直分析フィルタとを有し、画像データ群に対して、画面水平方向と画面垂直方向の両方について分析フィルタ処理を行う。ウェーブレット変換部410は、例えば、リフティング演算を用いてこのような分析フィルタ処理を行う。

【0460】

ウェーブレット変換部410は、途中計算用バッファ部411に格納された低域成分の係数データを再度読み出し、読み出した係数データに対して分析フィルタによるフィルタ処理を施して、高域成分および低域成分の係数のデータをさらに生成する。生成された係数データは、途中計算用バッファ部411に格納される。つまり、ウェーブレット変換部410は、低域成分に対して再帰的に分析フィルタ処理を繰り返し、係数データを成分毎に階層化する。

10

【0461】

ウェーブレット変換部410は、この処理を繰り返して分解レベルが所定レベルに達したら、途中計算用バッファ部411から係数データを読み出し、読み出された係数データを係数並び替え用バッファ部412に格納する。

【0462】

係数並び替え部413は、係数並び替え用バッファ部412に格納された係数データを、復号される順（ウェーブレット逆変換される順）に読み出し、エントロピ符号化部415に供給する。エントロピ符号化部415は、供給された係数データを、例えばハフマン符号化や算術符号化といった所定のエントロピ符号化方式で符号化する。

20

【0463】

エントロピ符号化部415は、レート制御部414と連動的に動作し、出力される圧縮符号化データのビットレートが略一定値となるように制御される。すなわち、レート制御部414は、エントロピ符号化部415からの符号化データ情報に基づき、エントロピ符号化部415により圧縮符号化されたデータのビットレートが目標値に達した時点あるいは目標値に達する直前でエントロピ符号化部415による符号化処理を終了するように制御する制御信号を、エントロピ符号化部415に対して供給する。エントロピ符号化部415は、レート制御部414から供給される制御信号に応じて符号化処理が終了した時点で、符号化データを出力する。

30

【0464】

画像符号化装置401は、このような画像の符号化を、画面の縦方向について、数ライン毎の処理に分割して、複数回に分けて段階的に行う。より具体的には、画像符号化装置401は、ウェーブレット変換後の最低域成分のサブバンド1ライン分の係数データを生成するために必要なライン数毎に、入力画像データを符号化する。

【0465】

以下において、この最低域成分のサブバンドの1ライン分の係数データを生成するために必要な、他のサブバンドも含めたラインの集まりを、ラインブロック（またはプレシント）と称する。ここでラインとは、ウェーブレット変換前の画像データに対応するピクチャ若しくはフィールド内、または各サブバンド内において形成される1行分の画素データ若しくは係数データのことを示す。

40

【0466】

なお、ラインブロック（プレシント）を構成するデータは、画像データであっても良いし、ウェーブレット変換された係数データであっても良いし、エントロピ符号化された符号化データであってもよい。つまり、ラインブロック（プレシント）は、ウェーブレット変換前の元の画像データにおける、ウェーブレット変換後の最低域成分のサブバンド1ライン分の係数データを生成するために必要なライン数分の画素データ群に対応するデータであれば、どのような状態のデータであってもよい。例えば、その画素データ群をウェーブレット変換して得られる各サブバンドの係数データ群のことを示す場合もある。

【0467】

50

つまり、ウェーブレット変換部 410 は、ラインブロック（プレシント）毎にウェーブレット変換処理を行う。すなわち、ウェーブレット変換部 410 は、最低域成分までの分析フィルタ処理が可能な最小量の入力画像データが途中計算用バッファ部 411 に蓄積され次第、ウェーブレット変換処理を開始する。

【0468】

従来のウェーブレット変換の方法の場合、まず、水平分析フィルタリング処理をピクチャ全体に対して行い、次に垂直分析フィルタリング処理をそのピクチャ全体に対して行う。そして得られた低域成分全体に対して同様の水平分析フィルタリング処理と垂直分析フィルタリング処理を順に行う。以上のように、分解レベルが最終レベルに達するまで、分析フィルタリング処理が再帰的に繰り返される。従って、各分析フィルタリング処理の結果をバッファに保持させる必要があるが、その際、バッファは、ピクチャ全体、若しくは、その時点の分解レベルの低域成分全体のフィルタリング結果を保持する必要があり、多大なメモリ容量を必要とすることになる（保持するデータ量が多い）。

【0469】

また、この場合、ピクチャ内において全てのウェーブレット変換が終了しないと、後段の係数並び替えやエントロピ符号化を行うことができず、遅延時間が増大する。

【0470】

これに対して、ウェーブレット変換部 410 は、上述したようにラインブロック単位で垂直分析フィルタリング処理および水平分析フィルタリング処理を最終レベルまで連続して行うので、従来の方法と比較して、一度に（同時期に）保持する（バッファリングする）必要のあるデータの量が少なく、用意すべきバッファのメモリ量を大幅に低減させることができる。また、最終レベルまで分析フィルタリング処理が行われることにより、後段の係数並び替えやエントロピ符号化等の処理も実行可能となる。

【0471】

係数並び替え部 413 は、次に読み出す係数データが係数並び替え用バッファ部 412 に格納され次第、その係数データを読み出す。エントロピ符号化部 415 は、供給される係数データを順次エントロピ符号化する。したがって、ウェーブレット変換部 410 がより低遅延に係数データの出力を開始することにより、係数並び替え部 413 やエントロピ符号化部 415 も、より低遅延に各自の処理を開始することができる。

【0472】

つまり、符号化データがより低遅延に出力されるようになる。以上のウェーブレット変換、係数並び替え、およびエントロピ符号化の各処理は、互いに並列に処理を行うことができる。従って、従来の方法と比較して遅延時間を大幅に低減させることができる。つまり、画像符号化装置 401 は、より低遅延に入力画像データを符号化し、符号化データを出力することができる。

【0473】

このような、画像符号化装置 401 に、本技術を適用してもよい。例えば、エントロピ符号化部 415 として、図 16 のエントロピ符号化部 23 を適用する。このようにすることにより、画像符号化装置 401 は、より低遅延に符号化処理を行うことができるだけでなく、より容易に符号化処理を行うことができる。

【0474】

つまり、画像符号化装置 401 は、符号化処理の負荷を低減させることができる。これにより、より高速に符号化処理を行うようにすることができる。また、より安価に画像符号化装置 401 を実現することができる（製造コストを低減させることができる）。さらに、画像符号化装置 401 は、符号化処理を、互いに並行して実行される複数のパイプライン処理として実行することができる。これにより、符号化処理のスループットを向上させることができる。さらに、画像符号化装置 401 は、ゼロラン部分をより効率よく符号化することができ、0 が多い係数データの符号化の符号化効率を向上させることができる。

【0475】

[画像復号装置]

図 4 0 は、画像復号装置の他の構成例を示すブロック図である。図 4 0 に示される画像復号装置 4 2 0 は、図 3 9の画像符号化装置 4 0 1 に対応する画像処理装置である。つまり、画像復号装置 4 2 0 は、画像符号化装置 4 0 1 が、画像データを符号化して生成した符号化データを復号し、復号画像データを生成する。

【 0 4 7 6 】

図 4 0 に示されるように、画像復号装置 4 2 0 は、エントロピ復号部 4 2 1、係数バッファ部 4 2 2、およびウェーブレット逆変換部 4 2 3 を有する。

【 0 4 7 7 】

画像復号装置 4 2 0 に入力された符号化データは、エントロピ復号部 4 2 1 に供給される。エントロピ復号部 4 2 1 は、その符号化データを、エントロピ符号化部 4 1 5 によるエントロピ符号化に対応する方法でエントロピ復号し、エントロピ符号化前の係数データを復元する。エントロピ復号部 4 2 1 は、その係数データを係数バッファ部 4 2 2 に供給し、格納させる。

【 0 4 7 8 】

ウェーブレット逆変換部 4 2 3 は、係数バッファ部 4 2 2 に格納された係数データを読み出し、垂直方向および水平方向にそれぞれ合成フィルタ処理を行い、その合成フィルタ処理の結果を再び係数バッファ部 4 2 2 に格納する。ウェーブレット逆変換部 4 2 3 は、この処理を分解レベルに応じて繰り返して、復号画像データ（出力画像データ）を得る。ウェーブレット逆変換部 4 2 3 は、生成した復号画像データを画像復号装置 4 2 0 の外部

【 0 4 7 9 】

このとき、係数データは、画像符号化装置 4 0 1 の係数並び替え部 4 1 3 により、ウェーブレット逆変換される順に並び替えられているので、ウェーブレット逆変換部 4 2 3 は、係数バッファ部 4 2 2 に格納される係数を順次読み出し、合成フィルタ処理に用いることができる。つまり、ウェーブレット逆変換部 4 2 3 は、より低遅延にウェーブレット逆変換を行うことができる。

【 0 4 8 0 】

したがって、画像復号装置 4 2 0 は、より低遅延に符号化データを復号し、復号画像データを出力することができる。

【 0 4 8 1 】

このような、画像復号装置 4 2 0 に、本技術を適用してもよい。例えば、エントロピ復号部 4 2 1 として、図 2 4のエントロピ復号部 1 2 1 を適用する。このようにすることにより、画像復号装置 4 2 0 は、より低遅延に復号処理を行うことができるだけでなく、より高速に復号処理を行うことができる。また、画像復号装置 4 2 0 は、より容易に復号処理を行うことができる。

【 0 4 8 2 】

つまり、画像復号装置 4 2 0 は、復号処理の負荷を低減させることができる。これにより、より高速に復号処理を行うようにすることができる。また、より安価に画像復号装置 4 2 0 を実現することができる（製造コストを低減させることができる）。さらに、画像復号装置 4 2 0 は、復号処理を、互いに並行して実行される複数のパイプライン処理として実行することができる。これにより、復号処理のスループットを向上させることができる。さらに、画像復号装置 4 2 0 は、ゼロラン部分の、より効率よい符号化を実現することができる。0 が多い係数データの符号化の符号化効率の向上を実現させることができる。

【 0 4 8 3 】

[符号化・復号の流れ]

以上のような画像符号化装置 4 0 1 および画像復号装置 4 2 0 を用いて、より低遅延にデータ伝送可能なデータ伝送システムを実現することができる。

【 0 4 8 4 】

データ伝送において、データを符号化して伝送し、伝送先において復号することは、伝

10

20

30

40

50

送効率の向上の為に効果的である。

【0485】

このように画像データを符号化して伝送するデータ伝送システムにおいて、伝送元の画像符号化装置として、画像符号化装置401を適用し、伝送先の画像復号装置として、画像復号装置420を適用する。このようにすることにより、上述したように、より低遅延な符号化・復号を実現することができるので、より低遅延なデータ伝送を実現することができる。

【0486】

より具体的に説明する。図41は、画像データを、画像符号化装置401によって符号化し、その符号化データを、画像復号装置420によって復号する流れの例を説明する図である。図41は、5×3フィルタを用いて、分解レベル=2までウェーブレット変換によるフィルタ処理を施した例である。ウェーブレット変換部410において、図41Aに一例が示されるように、入力画像データの第1ラインから第7ラインに対して1回目の分析フィルタ処理が水平および垂直方向にそれぞれ行われる(図41AのIn-1)。

10

【0487】

1回目の分析フィルタ処理の分解レベル=1の処理において、3ライン分の係数データが生成され、図41Bに一例が示されるように、分解レベル=1で形成される領域HH、領域HLおよび領域LHのそれぞれに配置される(図41BのWT-1)。

【0488】

また、分解レベル=1で形成される領域LLは、分解レベル=2による水平および垂直方向の分析フィルタ処理でさらに4分割される。分解レベル=2で生成される係数データは、分解レベル=1による領域LL内の、領域LL、領域HH、領域HL、および領域LHのそれぞれに、1ラインずつ配置される。

20

【0489】

ウェーブレット変換部410による2回目以降のフィルタ処理では、4ライン毎にフィルタ処理が行われ(図41AのIn-2・・・)、分解レベル=1で2ラインずつの係数データが生成され(図41BのWT-2)、分解レベル=2で1ラインずつの係数データが生成される。

【0490】

図41Bのようにウェーブレット変換されたデータを復号する際には、図41Cに一例が示されるように、符号化側の第1ライン乃至第7ラインによる1回目のフィルタ処理に対して、復号側の1回目の合成処理による第1ラインが出力される(図41CのOut-1)。以降、符号化側の2回目から最後の回の前までのフィルタ処理に対して、復号側で4ラインずつが出力される(図41CのOut-2・・・)。そして、符号化側の最後の回のフィルタ処理に対して、復号側で7ラインが出力される。

30

【0491】

以上のように、画像符号化装置401による符号化処理と、画像復号装置420による復号処理は、それぞれ、ラインブロック毎に行うことができ、また、互いに並行に処理を行うことができる。したがって、画像符号化装置401に画像データが入力されてから、画像復号装置420から復号画像データが出力されるまでの遅延時間が大幅に低減される。つまり、より低遅延なデータ伝送を実現することができる。

40

【0492】

このような、データ伝送システムの画像符号化装置401および画像復号装置420に、本技術を、上述したように適用することにより、より容易な符号化処理や復号処理を実現することができる。さらに、符号化処理や復号処理のスループットを向上させることができるので、データ伝送の遅延をさらに低減させることができる。さらに、0が多い係数データの符号化の符号化効率を向上させることができるので、伝送効率をさらに向上させることができる。

【0493】

なお、画像符号化装置401および画像復号装置420の構成は、本技術を適用するこ

50

とができる限り、上述した例に限らない。例えば、係数の並び替えは、画像復号装置 4 2 0 において行われるようにしてもよい。また、ウェーブレット変換処理（ウェーブレット逆変換処理）の代わりにその他の変換処理（逆変換処理）を適用するようにしてもよい。さらに、係数データに対して量子化・逆量子化を行うようにしてもよい。

【 0 4 9 4 】

以上のようなデータ伝送システムの具体例として、例えば、テレビジョン放送局や制作スタジオなどにおいて、スタジオ収録や中継などの際に、ビデオカメラと、カメラコントロールユニットやスイッチャとを接続する 1 本の同軸ケーブルで、映像信号、音声信号、送り返し（リターン）の映像信号、同期信号など複数の信号を重畳させて送信すると共に、電源の供給も行うようにしたデジタルトライアックスシステムがある。

10

【 0 4 9 5 】

また、このデータ伝送システムは、例えば、テレビジョン会議システム、家庭用ゲーム機器とビデオカメラやモニタ等の周辺機器との間の通信システム等、任意のシステムに適用することができる。さらに、このデータ伝送システムは、装置内部の処理部間のデータ伝送にも適用することができる。

【 0 4 9 6 】

すなわち、本技術は、画像データを符号化した符号化データを伝送するあらゆるシステムに適用することができる。

【 0 4 9 7 】

< 6 . 第 6 の実施の形態 >

20

[パーソナルコンピュータ]

上述した一連の処理は、ハードウェアにより実行させることもできるし、ソフトウェアにより実行させることもできる。この場合、例えば、図 4 2 に示されるようなパーソナルコンピュータとして構成されるようにしてもよい。

【 0 4 9 8 】

図 4 2 において、パーソナルコンピュータ 6 0 0 の CPU (Central Processing Unit) 6 0 1 は、ROM (Read Only Memory) 6 0 2 に記憶されているプログラム、または記憶部 6 1 3 から RAM (Random Access Memory) 6 0 3 にロードされたプログラムに従って各種の処理を実行する。RAM 6 0 3 にはまた、CPU 6 0 1 が各種の処理を実行する上において必要なデータなども適宜記憶される。

30

【 0 4 9 9 】

CPU 6 0 1、ROM 6 0 2、および RAM 6 0 3 は、バス 6 0 4 を介して相互に接続されている。このバス 6 0 4 にはまた、入出力インタフェース 6 1 0 も接続されている。

【 0 5 0 0 】

入出力インタフェース 6 1 0 には、キーボード、マウスなどよりなる入力部 6 1 1、CRT (Cathode Ray Tube) ディスプレイや LCD (Liquid Crystal Display) 等のディスプレイ、並びにスピーカなどよりなる出力部 6 1 2、フラッシュメモリ等 SSD (Solid State Drive) やハードディスクなどよりなる記憶部 6 1 3、有線 LAN (Local Area Network) や無線 LAN のインタフェースやモデムなどよりなる通信部 6 1 4 が接続されている。通信部 6 1 4 は、インターネットを含むネットワークを介しての通信処理を行う。

40

【 0 5 0 1 】

入出力インタフェース 6 1 0 にはまた、必要に応じてドライブ 6 1 5 が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブルメディア 6 2 1 が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 6 1 3 にインストールされる。

【 0 5 0 2 】

上述した一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、ネットワークや記録媒体からインストールされる。

【 0 5 0 3 】

この記録媒体は、例えば、図 4 2 に示されるように、装置本体とは別に、ユーザにプロ

50

グラムを配信するために配布される、プログラムが記録されている磁気ディスク（フレキシブルディスクを含む）、光ディスク（CD-ROM（Compact Disc - Read Only Memory）、DVD（Digital Versatile Disc）を含む）、光磁気ディスク（MD（Mini Disc）を含む）、若しくは半導体メモリなどよりなるリムーバブルメディア621により構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに配信される、プログラムが記録されているROM602や、記憶部613に含まれるハードディスクなどにより構成される。

【0504】

なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであっても良いし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであっても良い。

10

【0505】

また、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0506】

また、本明細書において、システムとは、複数のデバイス（装置）により構成される装置全体を表すものである。

【0507】

また、以上において、1つの装置（または処理部）として説明した構成が、複数の装置（または処理部）として構成されるようにしてもよい。逆に、以上において複数の装置（または処理部）として説明した構成が、まとめて1つの装置（または処理部）として構成されるようにしてもよい。また、各装置（または各処理部）の構成に上述した以外の構成が付加されるようにしてももちろんよい。さらに、システム全体としての構成や動作が実質的に同じであれば、ある装置（または処理部）の構成の一部が他の装置（または他の処理部）の構成に含まれるようにしてもよい。つまり、本開示の実施の形態は、上述した実施の形態に限定されるものではなく、本開示の要旨を逸脱しない範囲において種々の変更が可能である。

20

【0508】

なお、本技術は以下のような構成も取ることができる。

(1) 画像データから生成された複数の係数データを所定の数毎に1組とし、毎周期1組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化する有効桁数符号化部と、

30

値が0の係数データのみで構成される組により構成されるゼロランを、前記有効桁数符号化部の符号化と異なる周期において符号化するゼロラン符号化部と、

前記ゼロラン以外の係数データについて、前記有効桁数符号化部が符号化した組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数符号化部の符号化と異なる周期において符号化する絶対値符号化部と、

前記ゼロラン以外の係数データについて、前記絶対値符号化部が符号化した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値符号化部の符号化と異なる周期において符号化するサイン符号化部と

40

を備える画像処理装置。

(2) 前記ゼロラン符号化部は、前記ゼロランを構成する前記組の数の2進表現の桁数より1少ない数の符号0と、前記組の数の2進表現とにより構成される符号を生成する前記(1)に記載の画像処理装置。

(3) 前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より1少ない数の2進表現の桁数の符号0と、符号1とにより構成される符号を生成する

前記(2)に記載の画像処理装置。

(4) 前記ゼロラン符号化部は、前記ゼロランが処理対象のラインの最後まで続く場合、前記ゼロランを構成する前記組の数より1少ない数の2進表現の桁数の符号0により

50

構成される符号を生成する

前記(2)に記載の画像処理装置。

(5) 前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期において、前記ゼロランを符号化する

前記(1)乃至(4)のいずれかに記載の画像処理装置。

(6) 前記ゼロラン符号化部は、前記有効桁数符号化部が、前記ゼロランの後の組について前記最大有効桁数に関する情報を符号化する周期よりも前の周期において、前記ゼロランを符号化する

前記(1)乃至(5)のいずれかに記載の画像処理装置。

(7) 画像データをウェーブレット変換するウェーブレット変換部と、前記ウェーブレット変換部により画像データがウェーブレット変換されて得られたウェーブレット係数を量子化する量子化部と

をさらに備え、

前記有効桁数符号化部、前記ゼロラン符号化部、前記絶対値符号化部、および前記サイン符号化部は、それぞれ、前記量子化部により前記ウェーブレット係数が量子化されて得られた量子化係数について符号化を行う

前記(1)乃至(6)のいずれかに記載の画像処理装置。

(8) 画像処理装置の画像処理方法であって、

有効桁数符号化部が、画像データから生成された複数の係数データを所定の数毎に1組とし、毎周期1組ずつ、絶対値が最も大きい係数データの有効桁数である最大有効桁数を求め、前記最大有効桁数に関する情報を符号化し、

ゼロラン符号化部が、値が0の係数データのみで構成される組により構成されるゼロランを、前記有効桁数の符号化と異なる周期において符号化し、

絶対値符号化部が、前記ゼロラン以外の係数データについて、前記有効桁数の符号化が行われた組の各係数データの前記最大有効桁数分の絶対値を抽出し、前記絶対値を、前記有効桁数の符号化と異なる周期において符号化し、

サイン符号化部が、前記ゼロラン以外の係数データについて、前記絶対値が符号化された組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値の符号化と異なる周期において符号化する

画像処理方法。

(9) 複数の係数データを所定の数毎に1組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期1組ずつ、復号する有効桁数復号部と、

値が0の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数復号部の復号と異なる周期において復号するゼロラン復号部と、

前記ゼロラン以外の係数データについて、前記有効桁数復号部が復号した組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数復号部の復号と異なる周期において復号する絶対値復号部と、

前記ゼロラン以外の係数データについて、前記絶対値復号部が復号した組の前記絶対値が0でない各係数データの正負のサインを、前記絶対値復号部の復号と異なる周期において復号するサイン復号部と

を備える画像処理装置。

(10) 前記ゼロラン復号部は、前記有効桁数復号部による復号の結果、処理対象である今の組の最大有効桁数が0である場合、前記ゼロランを示す符号を復号する

前記(9)に記載の画像処理装置。

(11) 前記ゼロラン復号部は、符号1を読み込むか、ラインの最後まで符号を読み込むまで、順次符号を読み込む1回目の読み込みを行い、前記1回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達していない場合、前記1回目の読み込みで読み込んだ符号0の数分の符号をさらに読み込む2回目の読み込みを行い、前記2回

10

20

30

40

50

目の読み込みで読み込んだ符号の先頭に符号 1 をつけた符号列を 2 進表現とする数の、値が 0 の係数データを生成する

前記 (1 0) に記載の画像処理装置。

(1 2) 前記ゼロラン復号部は、前記 1 回目の読み込みで読み込んだ符号の示すゼロランが前記ラインの最後まで達した場合、前記ラインの最後まで達する数の、値が 0 の係数データを生成する

前記 (1 1) に記載の画像処理装置。

(1 3) 前記有効桁数復号部が前記最大有効桁数を示す符号を復号して得られた前記最大有効桁数を、所定の期間保持する最大有効桁数保持部をさらに備え、

前記絶対値復号部は、前記最大有効桁数保持部により保持される前記最大有効桁数分の前記絶対値を示す符号を、前記有効桁数復号部による復号が行われた周期から、前記最大有効桁数保持部が前記最大有効桁数を保持した期間分遅延した周期において、復号する

前記 (1 2) に記載の画像処理装置。

(1 4) 前記最大有効桁数保持部は、前記最大有効桁数を、次の周期まで保持する

前記 (1 3) に記載の画像処理装置。

(1 5) 前記最大有効桁数保持部は、前記最大有効桁数を、複数周期後の周期まで保持する

前記 (1 3) に記載の画像処理装置。

(1 6) 前記絶対値復号部による復号の結果得られた絶対値、および、前記サイン復号部による復号の結果得られたサインよりなる量子化係数を逆量子化する逆量子化部と、前記逆量子化部により前記量子化係数が逆量子化されて得られたウェーブレット係数を、ウェーブレット逆変換するウェーブレット逆変換部と

をさらに備える前記 (9) 乃至 (1 5) に記載の画像処理装置。

(1 7) 画像処理装置の画像処理方法であって、

有効桁数復号部が、複数の係数データを所定の数毎に 1 組とし、前記組において絶対値が最も大きい係数データの有効桁数である最大有効桁数を示す符号を、毎周期 1 組ずつ、復号し、

ゼロラン復号部が、値が 0 の係数データのみで構成される組により構成されるゼロランを示す符号を、前記有効桁数の復号と異なる周期において復号し、

絶対値復号部が、前記ゼロラン以外の係数データについて、前記有効桁数が復号された組の各係数データの前記最大有効桁数分の絶対値を示す符号を、前記有効桁数の復号と異なる周期において復号し、

サイン復号部が、前記ゼロラン以外の係数データについて、前記絶対値が復号された組の前記絶対値が 0 でない各係数データの正負のサインを、前記絶対値の復号と異なる周期において復号する

画像処理方法。

【符号の説明】

【 0 5 0 9 】

1 1 画像符号化装置, 2 3 エントロピ符号化部, 6 1 ライン判定部, 6 2 VLC符号化部, 6 3 最大有効桁数計算部, 6 4 VLC符号化部, 6 5 有効桁抽出部, 6 6 VLC符号化部, 6 7 サイン抽出部, 6 8 VLC符号化部, 6 9 符号連結部, 7 1 乃至 7 5 遅延部, 8 1 最大有効桁数変化符号化部, 8 2 最大有効桁数変化量符号化部, 8 3 0 連長符号化部, 1 1 1 画像復号装置, 1 2 1 エントロピ復号部, 1 5 1 符号分割部, 1 5 2 ライン判定部, 1 5 3 発生部, 1 5 4 乃至 1 5 6 VLC復号部, 1 5 7 量子化係数合成部, 1 5 8 切替部, 1 6 1 および 1 6 2 遅延部, 2 0 1 最大有効桁数変化復号部, 2 0 2 最大有効桁数変化量復号部, 2 0 3 0 連長復号部, 3 0 1 VLC符号化部, 3 0 2 遅延部, 3 5 1 VLC復号部

10

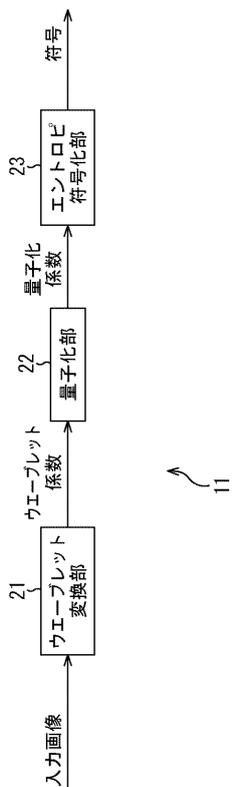
20

30

40

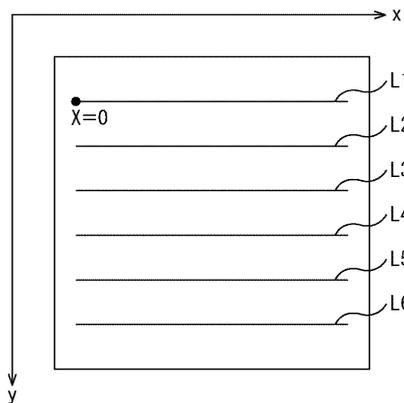
【 図 1 】

図1



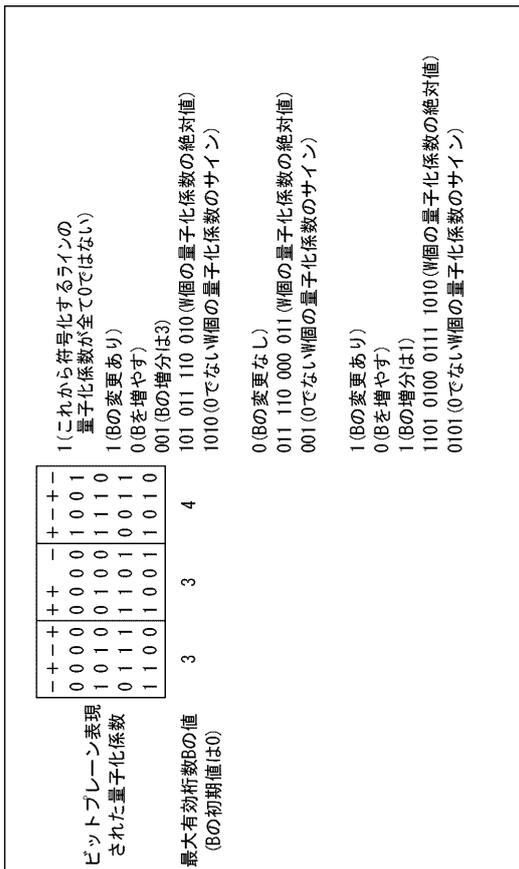
【 図 2 】

図2



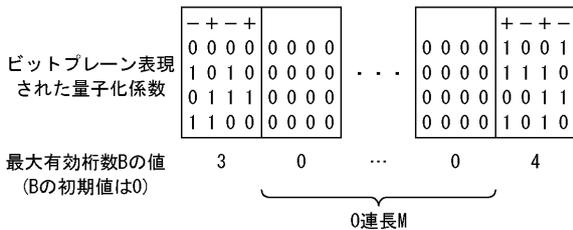
【 図 3 】

図3



【 図 4 】

図4



- 1 (これから符号化するラインの量子化係数が全て0ではない)
- 1 (Bの変更あり)
- 0 (Bを増やす)
- 001 (Bの増分は3)
- 101 011 110 010 (W個の量子化係数の絶対値)
- 1010 (0でないW個の量子化係数のサイン)
- 1 (Bの変更あり)
- 1 (Bを減らす)
- 00 (Bの減分は3)
- 00...0, M (0連長符号、カウント値の2進表現の桁数と値)
- 0001 (Bの増分は4)
- 1101 0100 0111 1010 (W個の量子化係数の絶対値)
- 0101 (0でないW個の量子化係数のサイン)

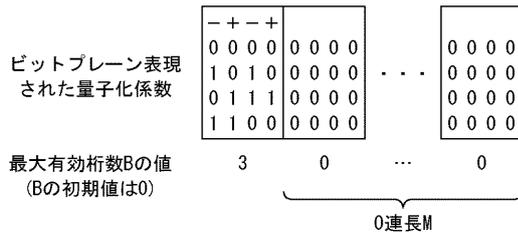
【図 5】

図5

0連長	2進表現	符号
1	1	1
2	10	01, 0
3	11	01, 1
4	100	001, 00
5	101	001, 01
6	110	001, 10
7	111	001, 11
8	1000	0001, 000
⋮	⋮	⋮
15	1111	0001, 111
⋮	⋮	⋮

【図 6】

図6



1 (これから符号化するラインの量子化係数が全て0ではない)

1 (Bの変更あり)
 0 (Bを増やす)
 001 (Bの増分は3)
 101 011 110 010 (W個の量子化係数の絶対値)
 1010 (0でないW個の量子化係数のサイン)

1 (Bの変更あり)
 1 (Bを減らす)
 00 (Bの減分は3)

00...01 (0連長符号 (終端)、(カウント値-1)の2進表現の桁数と1)

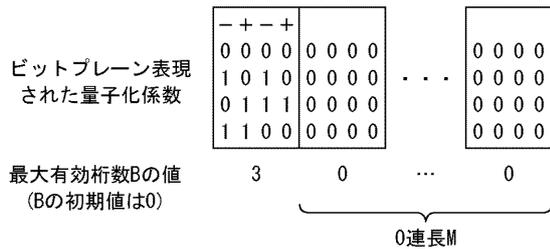
【図 7】

図7

0連長	2進表現	符号	備考
1	1	なし	B=0がラインの最後まで0確定を意味する
2	10	01	2以上のゼロランを表す符号のprefix
3	11	001	4以上のゼロランを表す符号のprefix
4	100	001	4以上のゼロランを表す符号のprefix
5	101	0001	8以上のゼロランを表す符号のprefix
6	110	0001	8以上のゼロランを表す符号のprefix
7	111	0001	8以上のゼロランを表す符号のprefix
8	1000	0001	8以上のゼロランを表す符号のprefix
9	1001	00001	16以上のゼロランを表す符号のprefix
10	1010	00001	16以上のゼロランを表す符号のprefix
11	1011	00001	16以上のゼロランを表す符号のprefix
12	1100	00001	16以上のゼロランを表す符号のprefix
13	1101	00001	16以上のゼロランを表す符号のprefix
14	1110	00001	16以上のゼロランを表す符号のprefix
15	1111	00001	16以上のゼロランを表す符号のprefix
16	10000	00001	16以上のゼロランを表す符号のprefix

【図 8】

図8



1 (これから符号化するラインの量子化係数が全て0ではない)

1 (Bの変更あり)
 0 (Bを増やす)
 001 (Bの増分は3)
 101 011 110 010 (W個の量子化係数の絶対値)
 1010 (0でないW個の量子化係数のサイン)

1 (Bの変更あり)
 1 (Bを減らす)
 00 (Bの減分は3)

00...0 (0連長符号 (終端)、(カウント値-1)の2進表現の桁数)

【 図 9 】

0連長	2進表現	符号	備考
1	1	なし	B=0がラインの最後まで0確定を意味する
2	10	0	2以上のゼロランを表す符号のprefix
3	11	00	4以上のゼロランを表す符号のprefix
4	100	00	4以上のゼロランを表す符号のprefix
5	101	000	8以上のゼロランを表す符号のprefix
6	110	000	8以上のゼロランを表す符号のprefix
7	111	000	8以上のゼロランを表す符号のprefix
8	1000	000	8以上のゼロランを表す符号のprefix
9	1001	0000	16以上のゼロランを表す符号のprefix
10	1010	0000	16以上のゼロランを表す符号のprefix
11	1011	0000	16以上のゼロランを表す符号のprefix
12	1100	0000	16以上のゼロランを表す符号のprefix
13	1101	0000	16以上のゼロランを表す符号のprefix
14	1110	0000	16以上のゼロランを表す符号のprefix
15	1111	0000	16以上のゼロランを表す符号のprefix
16	10000	0000	16以上のゼロランを表す符号のprefix

【 図 1 1 】

非pipelineでの符号順序

cycleと符号の関係
 cycle3: B(3) A(3) S(3)
 cycle4: B(4) A(4) S(4)
 cycle5: B(5) A(5) S(5)
 cycle6: B(6) A(6) S(6)

A

B(4)、B(5)でB=0の場合
 cycle3: B(3) A(3) S(3)
 cycle4: B(4) - -
 cycle5: B(5) - -
 cycle6: B(6) A(6) S(6)

B

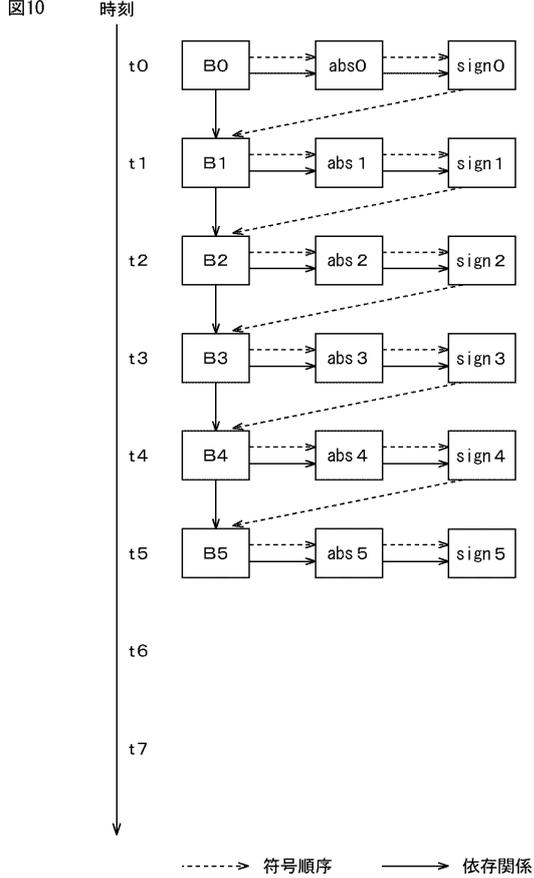
符号化の場合、0連長が確定するcycle6でB(4)B(5)B(6)の符号を出力する。
 cycle3: B(3) A(3) S(3)
 cycle4: - - -
 cycle5: - - -
 cycle6: B(4)B(5)B(6) A(6) S(6)

C

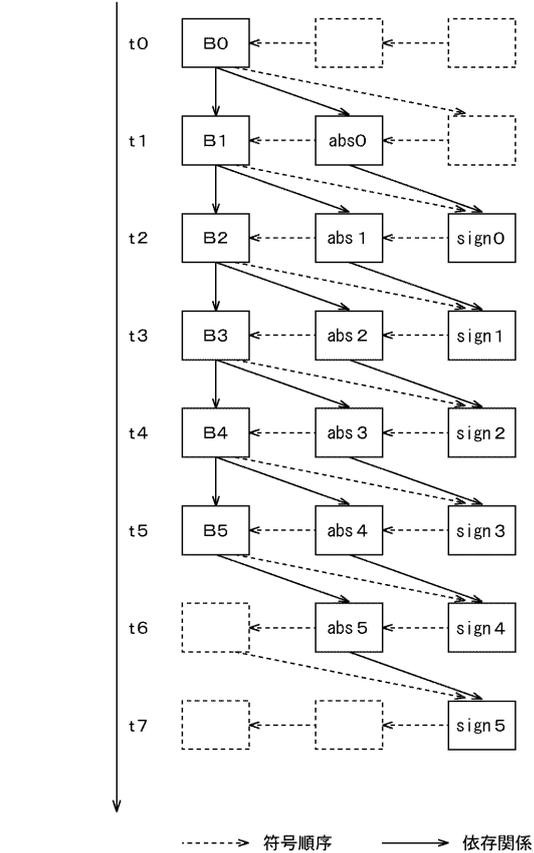
復号の場合、cycle4でB(4)~B(6)でのBの値を復号する。
 cycle3: B(3) A(3) S(3)
 cycle4: B(4)B(5)B(6) - -
 cycle5: - - -
 cycle6: - A(6) S(6)

D

【 図 1 0 】



【 図 1 2 】



【図 13】

図13

pipelineでの符号順序 (改善前)

cycleと符号の関係
 cycle4: S(2) A(3) B(4)
 cycle5: S(3) A(4) B(5)
 cycle6: S(4) A(5) B(6)
 cycle7: S(5) A(6) B(7)

A

B(4)、B(5)でB=0のとき

cycle4: S(2) A(3) B(4)
 cycle5: S(3) - B(5)
 cycle6: - B(6)
 cycle7: - A(6) B(7)

B

符号化の場合、0連長が確定するcycle6でB(4)B(5)B(6)の符号を出力する。

cycle4: S(2) A(3) -
 cycle5: S(3) - -
 cycle6: - B(4)B(5)B(6)
 cycle7: - A(6) B(7)

C

復号の場合、cycle4でB(4)~B(6)でのBの値を復号する。

cycle4: S(2) A(3) B(4)B(5)B(6)
 cycle5: S(3) - -
 cycle6: - A(6) B(7)
 cycle7: - A(6) B(7)

D

【図 15】

図15

pipelineでの符号順序 (改善後)

B(4)B(5)B(6)の符号を次のように分割して置くことにする。

cycle4: S(2) A(3) diff, minus, more*
 cycle5: S(3) - 0連長符号, more*, enough
 cycle6: - 0連長符号, more*, enough
 cycle7: - A(6) B(7)

A

符号化の場合、cycle4でB(4)の値を、cycle6でB(5)、B(6)の値を符号化する。

cycle4: S(2) A(3) diff, minus, more*
 cycle5: S(3) - 0連長符号, more*, enough
 cycle6: - 0連長符号, more*, enough
 cycle7: - A(6) B(7)

B

復号の場合、cycle4でB(4)の値を、cycle5でB(5)、B(6)でのBの値を復号する。

cycle4: S(2) A(3) diff, minus, more*
 cycle5: S(3) - 0連長符号, more*, enough
 cycle6: - 0連長符号, more*, enough
 cycle7: - A(6) B(7)

C

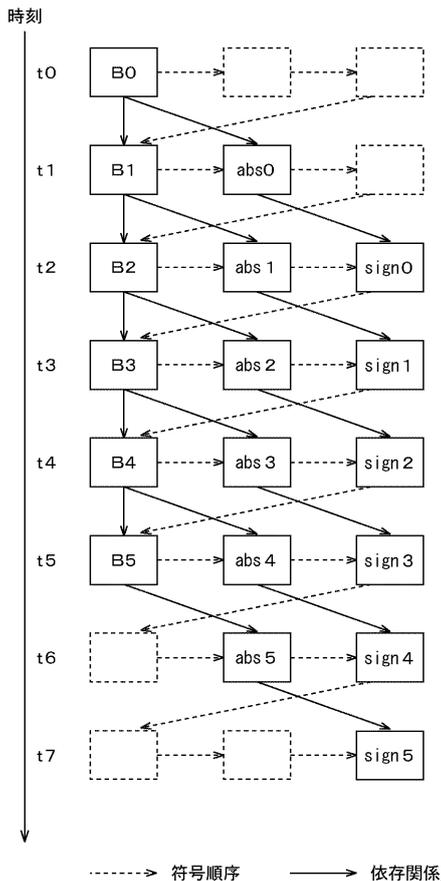
0連長符号, more*, enoughは連続しているので、cycle5~6への復号処理の配分には次などの自由度がある。

cycle4: S(2) A(3) diff, minus, more*
 cycle5: S(3) - 0連長符号
 cycle6: - more*, enough
 cycle7: - A(6) B(7)

D

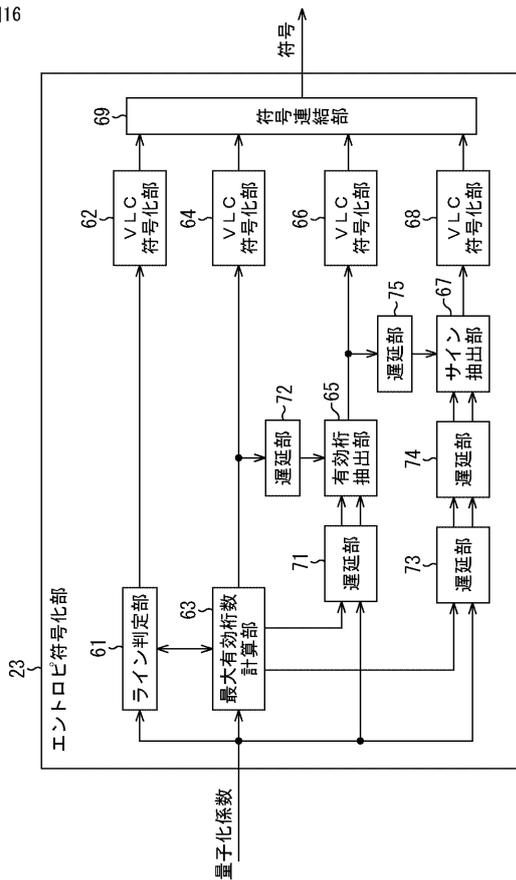
【図 14】

図14



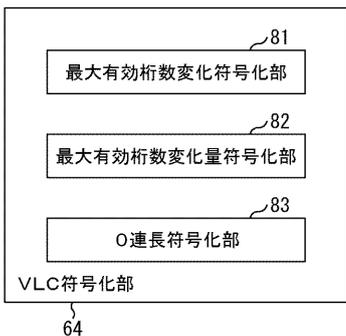
【図 16】

図16



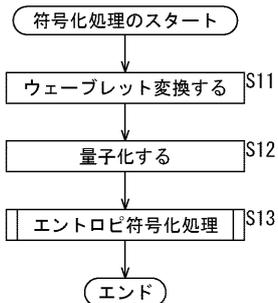
【図17】

図17



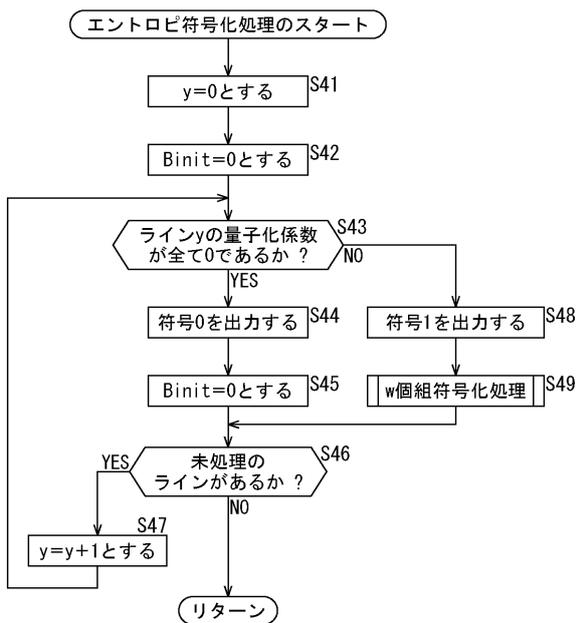
【図18】

図18



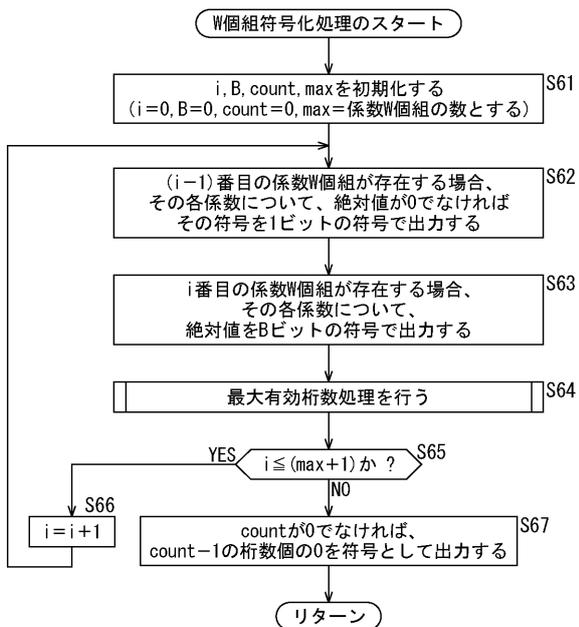
【図19】

図19



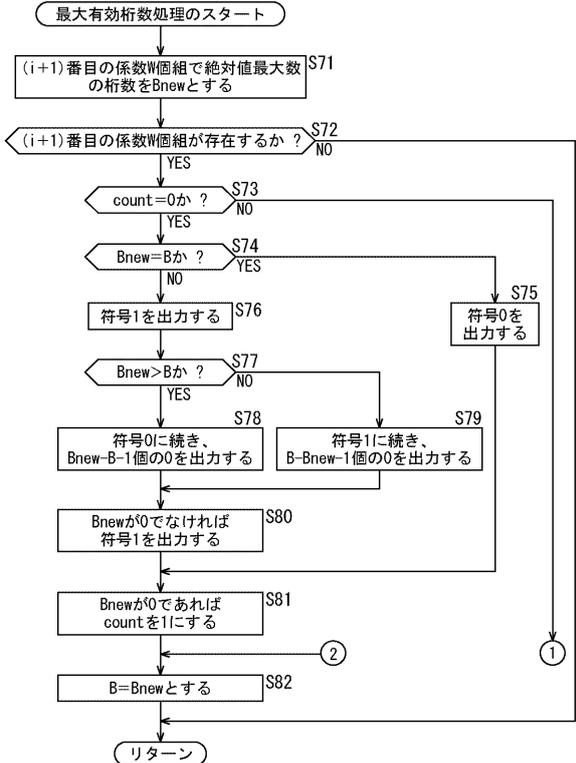
【図20】

図20



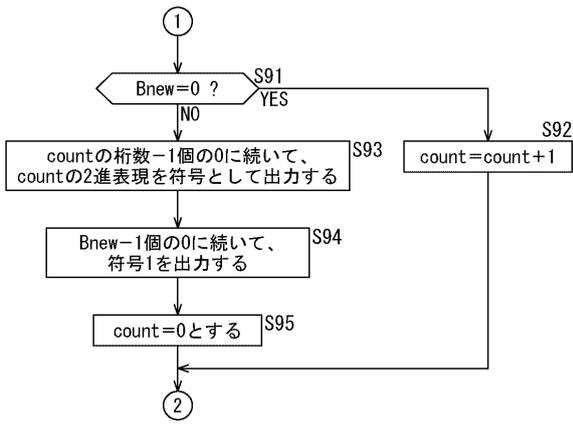
【図21】

図21



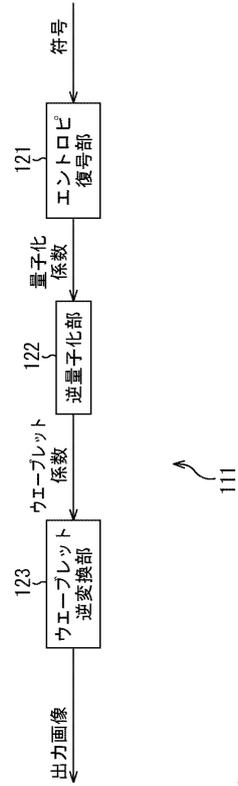
【図22】

図22



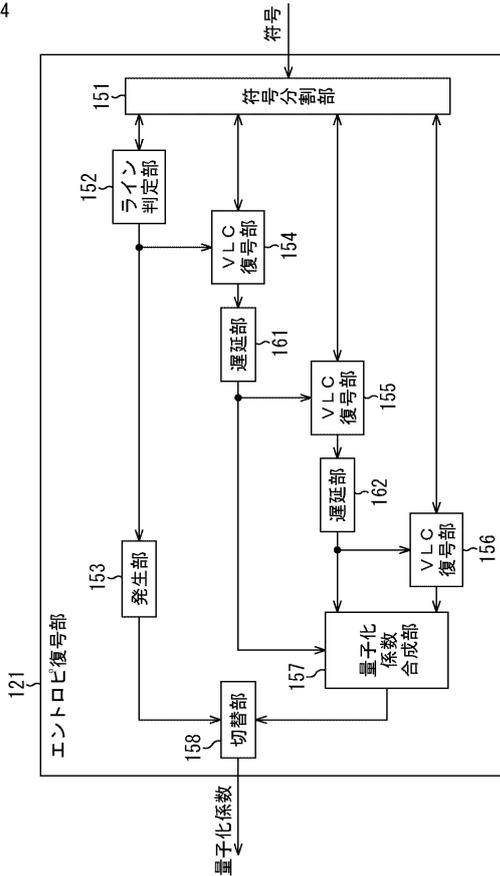
【図23】

図23



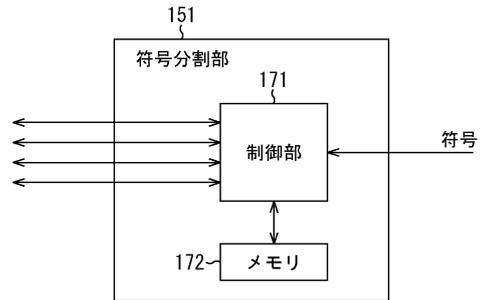
【図24】

図24



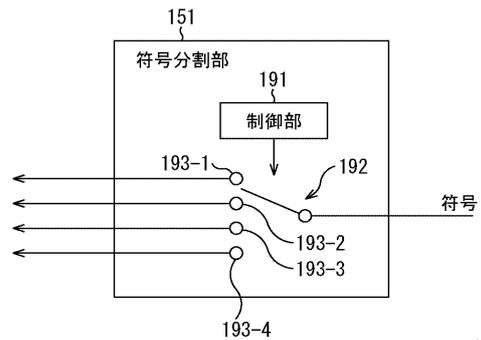
【図25】

図25



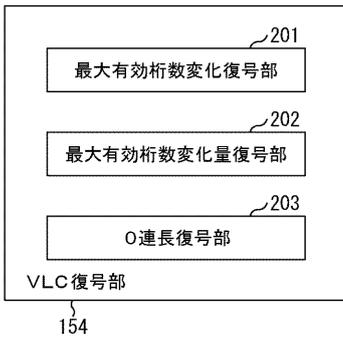
【図26】

図26



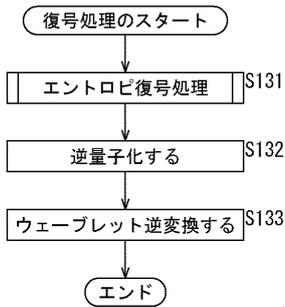
【図27】

図27



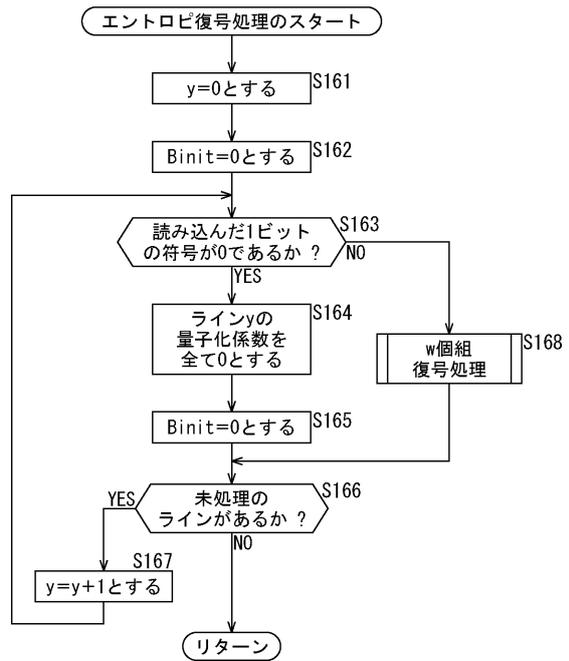
【図28】

図28



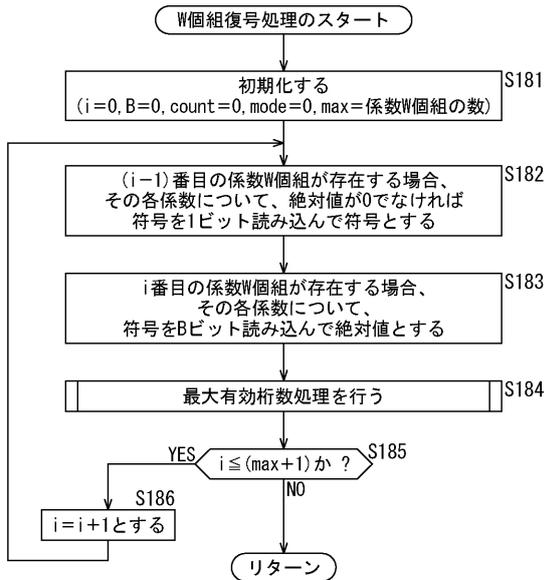
【図29】

図29



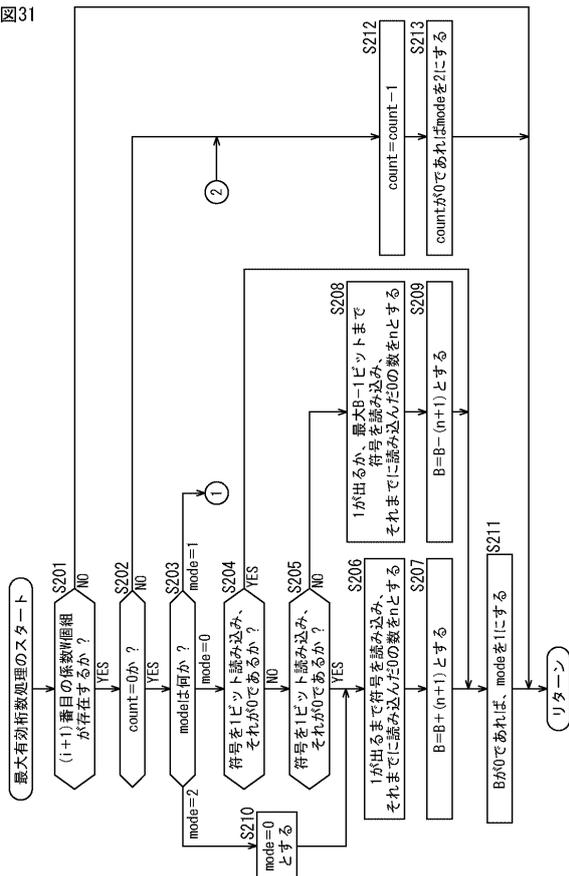
【図30】

図30



【図31】

図31



【図 36】

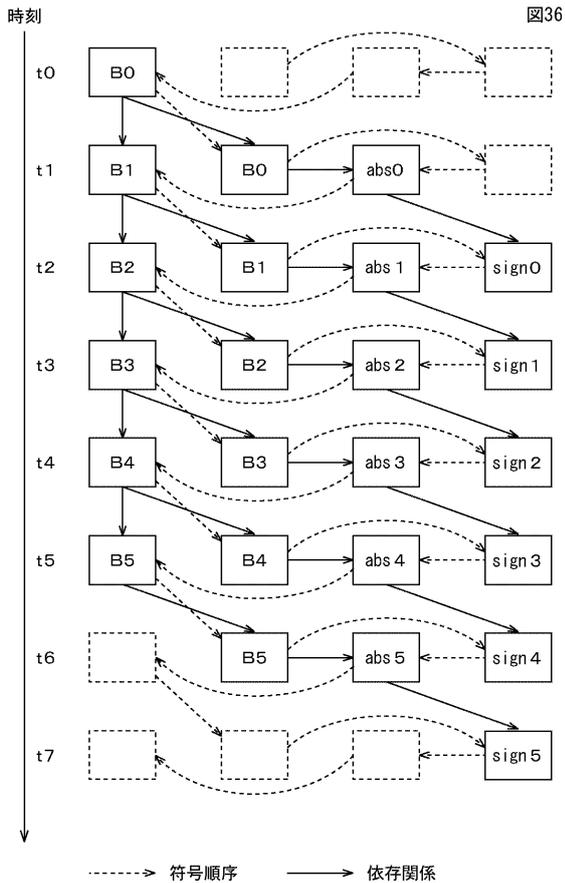


図36

【図 37】

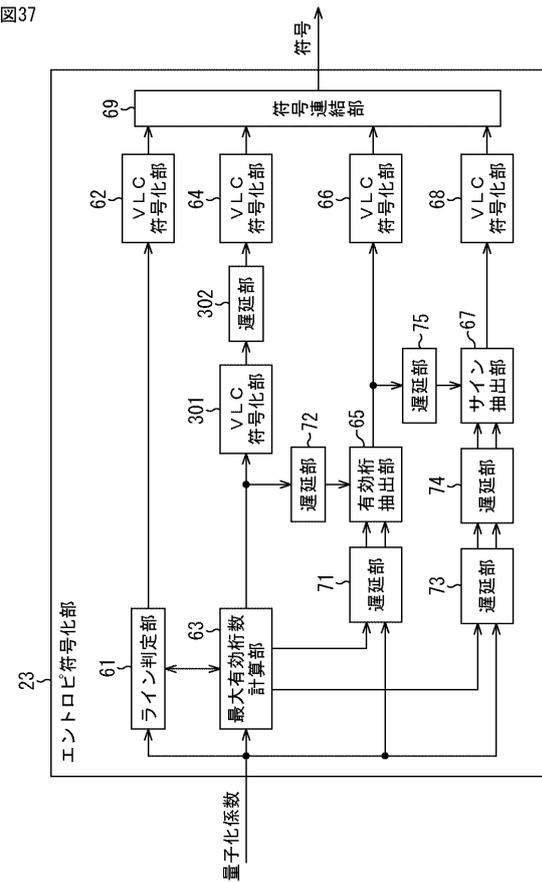


図37

【図 38】

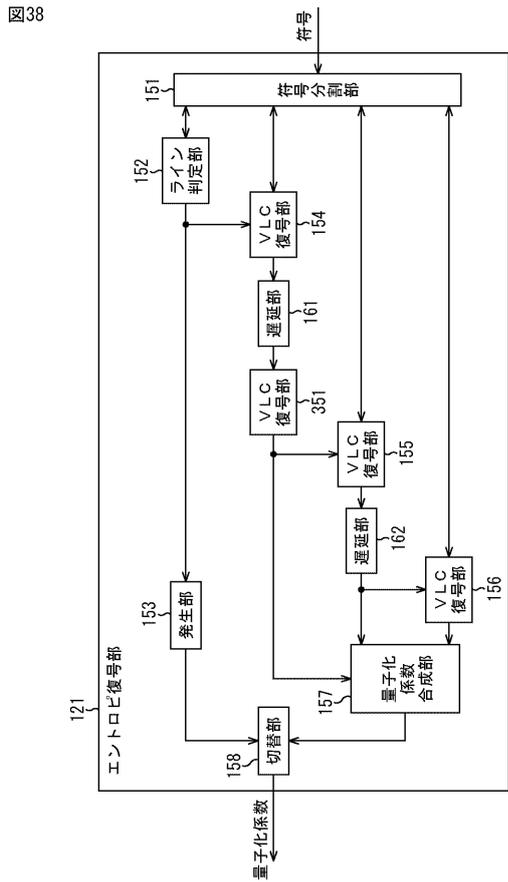


図38

【図 39】

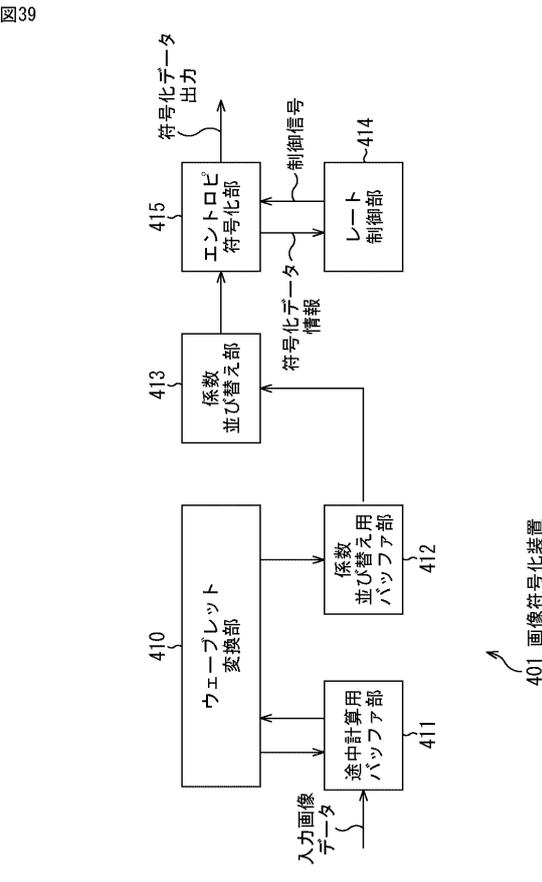
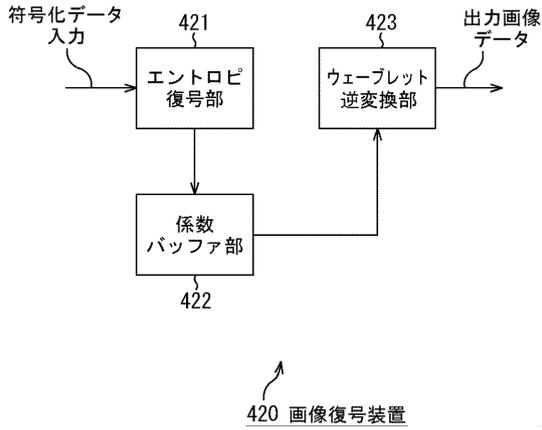


図39

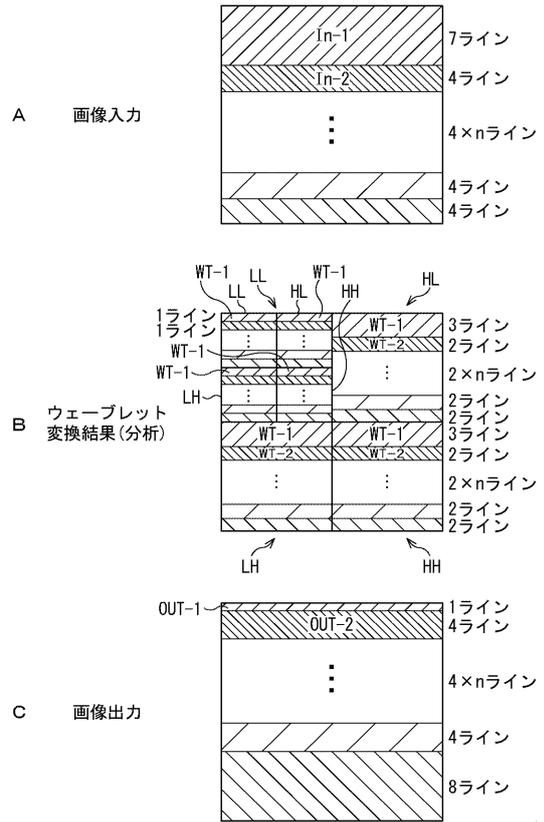
【図40】

図40



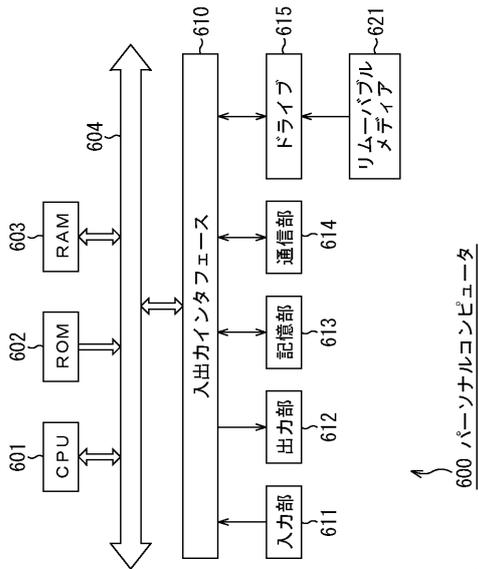
【図41】

図41



【図42】

図42



フロントページの続き

(56)参考文献 再公表特許第2007/058296(JP, A1)

特開昭59-044175(JP, A)

特開2006-157678(JP, A)

特開平11-317783(JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N 1/41

H03M 7/40

H04N 19/93