

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-86579  
(P2006-86579A)

(43) 公開日 平成18年3月30日(2006.3.30)

(51) Int. Cl.	F I	テーマコード (参考)
<b>HO4N 1/41 (2006.01)</b>	HO4N 1/41 B	5B057
<b>GO6T 9/00 (2006.01)</b>	GO6T 9/00	5C059
<b>HO4N 7/26 (2006.01)</b>	HO4N 7/13 Z	5C078

審査請求 未請求 請求項の数 11 O L (全 18 頁)

(21) 出願番号	特願2004-266653 (P2004-266653)	(71) 出願人	000006747 株式会社リコー
(22) 出願日	平成16年9月14日 (2004.9.14)		東京都大田区中馬込1丁目3番6号
		(74) 代理人	100101177 弁理士 柏木 慎史
		(74) 代理人	100072110 弁理士 柏木 明
		(72) 発明者	野水 泰之 東京都大田区中馬込1丁目3番6号 株式 会社リコー内
		Fターム(参考)	5B057 CA08 CA12 CA16 CB18 CC02 CG05 CG09 5C059 KK03 KK04 KK36 MA00 MA24 MC11 MC38 ME01 PP01 PP15 PP22 PP24 SS28
			最終頁に続く

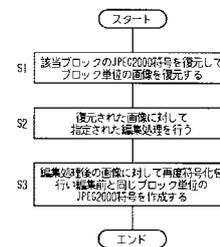
(54) 【発明の名称】 画像処理装置、プログラム、及び記憶媒体

(57) 【要約】

【課題】 画像データを圧縮符号化した符号列中で、ブロック単位の符号データを編集処理して、編集後の符号データと編集処理が施されていない符号データとを結合して新たな符号列を生成する処理を、高速に実現できるようにする。

【解決手段】 該当符号を復号してブロック単位の符号を復元し(ステップS1)、続いて、復元された画像に対して編集処理を行う(ステップS2)。最後に、編集後の画像に対して元の符号と同じ方法で符号化を行い、元の符号列と同じ形式の符号列を作成する(ステップS3)。ステップS3では、編集で作成された新たなブロック単位の符号データと符号列中の編集処理が施されていない符号データとを結合して新たな符号列を生成する。この場合、編集の前後のブロック単位の符号データの符号量に変化に応じて結合を実行する最適な手段を複数の手段の中から選択する。

【選択図】 図18



**【特許請求の範囲】****【請求項 1】**

元画像のデータを所定のフォーマットにより圧縮符号化した符号列中で独立して編集処理が可能な特定のブロック単位の符号データを対象に所定の編集処理を施して新たなブロック単位の符号データを作成する手段と、

前記符号列中の元のブロック単位の符号データを参照して、作成された前記新たなブロック単位の符号データと前記符号列中の前記編集処理が施されていない符号データとを結合して新たな符号列を生成する手段と、  
を備えている画像処理装置。

**【請求項 2】**

前記符号列を生成する手段は、前記編集の前後の前記ブロック単位の符号データの符号量に変化に応じて前記結合を実行する手段を複数の手段の中から選択する、請求項 1 に記載の画像処理装置。

10

**【請求項 3】**

前記符号列を生成する手段は、前記編集後の前記ブロック単位の符号データが前記編集前よりデータ量が多くなる場合には、前記新たに作成された符号データを前記新たな符号列の最後尾に追加する形で前記結合を行う、請求項 2 に記載の画像処理装置。

**【請求項 4】**

前記符号列を生成する手段は、前記編集後の前記ブロック単位の符号データが前記編集前よりデータ量が多くなる場合には、符号化パラメータの変更により前記編集後の前記ブロック単位の符号データが前記編集前の符号データのデータ量を超えないようにする、請求項 2 に記載の画像処理装置。

20

**【請求項 5】**

前記所定のフォーマットは J P E G 2 0 0 0 である、請求項 1 ~ 4 のいずれかの一に記載の画像処理装置。

**【請求項 6】**

元画像のデータを所定のフォーマットにより圧縮符号化した符号列中で独立して編集処理が可能な特定のブロック単位の符号データを対象に所定の編集処理を施して新たなブロック単位の符号データを作成する手段と、

前記符号列中の元のブロック単位の符号データを参照して、作成された前記新たなブロック単位の符号データと前記符号列中の前記編集処理が施されていない符号データとを結合して新たな符号列を生成する手段と、  
をコンピュータに実行させるコンピュータに読取可能なプログラム。

30

**【請求項 7】**

前記符号列を生成する手段は、前記編集の前後の前記ブロック単位の符号データの符号量に変化に応じて前記結合を実行する手段を複数の手段の中から選択する、請求項 6 に記載のプログラム。

**【請求項 8】**

前記符号列を生成する手段は、前記編集後の前記ブロック単位の符号データが前記編集前よりデータ量が多くなる場合には、前記新たに作成された符号データを前記新たな符号列の最後尾に追加する形で前記結合を行う、請求項 7 に記載のプログラム。

40

**【請求項 9】**

前記符号列を生成する手段は、前記編集後の前記ブロック単位の符号データが前記編集前よりデータ量が多くなる場合には、符号化パラメータの変更により前記編集後の前記ブロック単位の符号データが前記編集前の符号データのデータ量を超えないようにする、請求項 7 に記載のプログラム。

**【請求項 10】**

前記所定のフォーマットは J P E G 2 0 0 0 である、請求項 6 ~ 9 のいずれかの一に記載のプログラム。

**【請求項 11】**

50

請求項 6 ~ 10 のいずれかの一に記載のプログラムを記憶している、記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像データを圧縮符号化した符号列を編集する画像処理装置、この画像処理装置を実現するプログラム、及び、このプログラムを記憶している記憶媒体に関する。

【背景技術】

【0002】

多値画像(特にカラー画像)を可逆に符号化する符号化方式として、ITU-TとISO 10

とで標準方式とされているJPEG2000方式が知られている(非特許文献1を参照)

【0003】

【非特許文献1】ISO 15444-1 JPEG2000 Image Coding System (JPEG2000)

【発明の開示】

【発明が解決しようとする課題】

【0004】

近年のデジタル機器においては、高画質化のために、解像度を高くし、あるいは、階調 20  
数を多くする傾向がある。これにより画像の持つ情報量が多くなることで画質は向上する  
反面、画像の情報量が多くなるという問題がある。後者の例を挙げると、例えば、従来2  
階調(白または黒)値であった画像を白黒256階調の画像にすると、情報量は8倍にな  
ってしまう。情報量が8倍になるということは、その画像データを記憶するために必要と  
される記憶容量も単純に計算すると8倍になってしまい、装置の製造コストが増大する  
という問題になる。そこで、通常は記憶容量を削減するために、画像を圧縮符号化する。

【0005】

このような符号化方式の1つに、多階調画像を効率良く符号化するための技術が存在す 30  
る。この多階調画像(カラー画像も含む)の符号化方式の代表としては、ISOとITU  
-Tとで標準勧告されているJPEG方式がある。JPEG方式は、基本であるDCT方  
式とオプションのDPCMを用いた方式がある。前者は人間の視覚特性を利用して画質を  
損なわない程度に原画の情報量を一部削減して符号化を行う符号化方式(非可逆符号化 30  
方式、ロッキー(lossy)符号化方式と呼ばれる)であり、後者は原画の情報量を損なうこ  
となく符号化を行う符号化方式(可逆符号化方式、ロスレス(lossless)符号化方式と呼  
ばれる)である。

【0006】

DCT方式は、離散コサイン変換を使って画像情報を周波数情報に変換した後に情報の 40  
符号化を行う方式である。一方、DPCM方式は注目画素レベルを周囲画素より予測を行  
い、その予測誤差を符号化する方式である。画質重視で符号化を行うのであれば、効率の  
良いDCT方式を用いるのが良いが、情報の保存性という点ではDCT方式は非可逆であ  
るために、可逆であるDPCM方式となる。理想としては、可逆で高能率な方式があれば  
よいが、現状のDPCMによる可逆方式ではそれほど大きな効率を得られないという問題 40  
があり、パーソナルコンピュータ(PC)等で使用される比較的階調数の多い多値画像の  
圧縮には、DCT方式を使うことが主流になっている。しかし、DCT方式は圧縮率を高  
くすると特有のブロック歪みや輪郭部でモスキートノイズが発生し、画質が極端に劣化す  
る。特に文字画像において、その傾向が顕著であるために画質的に大きな問題となってい  
る。

【0007】

また、JPEG方式は、画像の記憶容量を少なくする用途では最適な方式であるが、デ 50  
ジタル複写機で使われる画像の編集・加工等の用途には最適ではない。なぜなら、符号状  
態で画像の位置を特定できない、言い換えれば、指定された画像の任意部分のみ復号処理  
することができないからである。よって、編集・加工処理を行うためには、一度、画像全

てを復号し、復号後の画像に対して編集・加工を行い、必要であれば、再度、符号化を行うということになり、復号後の画像を記憶するための大きな記憶容量のメモリが必要になるという問題がある（例えば、A4サイズ、600dpi、RGBカラー画像で、約100Mbyte必要である）。

**【0008】**

このような編集・加工処理時のメモリの記憶容量の問題を解決する手段の一つに、固定長の符号化方式を利用することが考えられる。画像の符号化には符号化後の符号語長から可変長と固定長に大きく分けられる。前者の特徴は、後者に比べて符号化効率が良い点と可逆も可能である点にある。これに対し、後者の特徴は、符号の状態ですべての画像の位置がわかるために、画像中の任意の部分のみを再生することなどが可能である。これは、符号状態のまま、画像の編集・加工処理等が可能になることを意味している。しかし、その反面、可変長符号に比べて、一般的に符号化効率が悪く、可逆符号化も困難であるという問題がある。

10

**【0009】**

以上のJPEG方式の欠点を解決するために、JPEG2000と称する符号化方式が近年、注目されている。JPEG2000は、ウェーブレット変換を用いた変換符号化方式で、今後、カラー画像をはじめとする静止画像の分野において、JPEGに置き換わっていくだろうと予測されている。JPEG2000は、JPEGの欠点である低ビットレートでの画質劣化を少なくしたことに加え、実用的な新機能を多数備えている。その中の機能にタイル処理というものがあり、これは画像を小さな領域に分けて独立に符号化を行うため、符号状態で画像の領域を特定することが可能になり、結果的に、符号状態のまま画像の編集・加工処理が可能になる。

20

**【0010】**

しかし、JPEG2000の技術にも、まだ問題はある。それは、符号列中の一部のフラグデータを編集処理した際に、その編集処理後のフラグデータが編集前のフラグデータよりデータ量が多くなってしまった場合である。この場合、編集後のフラグデータを符号列中に挿入するために、編集した部分以降のフラグデータを挿入するのに必要とされる分だけ、符号列を再配置する処理が必要となる。メモリ上での処理を例にとれば、フラグデータを挿入するのに必要なメモリ領域を確保するために、データの移動が必要になるということである。JPEG2000は、多機能かつ高性能のため処理が複雑になっており、JPEGとの比較を例にすると、ソフトウェアの処理では約4～5倍の処理時間を必要とする。このことを合わせて考えると、特に、上記のような編集作業を行う場合、処理時間が長くなり、ユーザに与える操作性の面で、大きな問題となってしまう。

30

**【0011】**

特に、JPEG2000符号化処理の手段の一つにタイル処理というものがある。これは画像をタイルと呼ばれる小領域に分割し、分割された領域単位で独立に符号化を行う処理である。この方法は符号列の編集処理等、画像の特定領域にランダムにアクセスしたいようなアプリケーションに最適である。特にJPEG2000は高効率の符号化方式を採用しているため、非常に複雑な演算を必要とする。良く比較されるJPEG方式と比べて、複雑度は数倍以上である。処理の複雑さはそのまま処理時間に比例するため、JPEG2000の処理はJPEGに比べて長くなるが、タイルを使うことで特定部分のみの符号処理で済むので処理時間の問題はなくなる。

40

**【0012】**

しかし、前述したように、編集後のフラグデータによっては全体の符号列を調整する必要があるという問題が残っている。

**【0013】**

本発明の目的は、画像データを圧縮符号化した符号列中で、ブロック単位のフラグデータを編集処理して、編集後のフラグデータと編集処理が施されていないフラグデータとを結合して新たな符号列を生成する処理を、高速に実現できるようにすることである。

**【課題を解決するための手段】**

50

## 【 0 0 1 4 】

本発明は、元画像のデータを所定のフォーマットにより圧縮符号化した符号列中で独立して編集処理が可能な特定のブロック単位の符号データを対象に所定の編集処理を施して新たなブロック単位の符号データを作成する手段と、前記符号列中の元のブロック単位の符号データを参照して、作成された前記新たなブロック単位の符号データと前記符号列中の前記編集処理が施されていない符号データとを結合して新たな符号列を生成する手段と、

を備えている画像処理装置である。

## 【 0 0 1 5 】

別の面から見た本発明は、元画像のデータを所定のフォーマットにより圧縮符号化した符号列中で独立して編集処理が可能な特定のブロック単位の符号データを対象に所定の編集処理を施して新たなブロック単位の符号データを作成する手段と、前記符号列中の元のブロック単位の符号データを参照して、作成された前記新たなブロック単位の符号データと前記符号列中の前記編集処理が施されていない符号データとを結合して新たな符号列を生成する手段と、をコンピュータに実行させるコンピュータに読取可能なプログラムである。

10

## 【 発明の効果 】

## 【 0 0 1 6 】

本発明によれば、符号列中の元のブロック単位の符号データを参照して、作成された新たなブロック単位の符号データと符号列中の編集処理が施されていない符号データとを結合して新たな符号列を生成するので、編集後の新たな符号データ列を編集処理が施されていない符号データと結合する手段を、編集前の符号列を参照しながら最短処理時間で行えるように全体符号列を制御するので、高速な編集処理が実現できる。

20

## 【 発明を実施するための最良の形態 】

## 【 0 0 1 7 】

[ J P E G 2 0 0 0 の概要について ]

最初に J P E G 2 0 0 0 の概要について説明する。

## 【 0 0 1 8 】

図 1 は、 J P E G 2 0 0 0 アルゴリズムの基本を説明するための機能ブロック図である。図 1 に示すように、 J P E G 2 0 0 0 アルゴリズムは、色空間変換・逆変換部 1 0 1、2 次元ウェーブレット変換・逆変換部 1 0 2、量子化・逆量子化部 1 0 3、エントロピー符号化・復号化部 1 0 4、タグ処理部 1 0 5 によって構成されている。以下、各部について説明する。

30

## 【 0 0 1 9 】

色空間変換・逆変換部 1 0 1 及び 2 次元ウェーブレット変換・逆変換部 1 0 2 について図 2 及び図 3 を参照しながら説明する。

## 【 0 0 2 0 】

図 2 は、カラー画像である原画像の分割された各コンポーネントの一例を示す模式図である。カラー画像は、一般に、図 2 に示すように、原画像の各コンポーネント R、G、B ( 1 1 1 ) が、例えば R G B 原色系によって分離されている。そして、原画像の各コンポーネント R、G、B は、さらに、矩形をした領域であるタイル 1 1 2 によって分割される。個々のタイル 1 1 2、例えば、R 0 0, R 0 1, ..., R 1 5 / G 0 0, G 0 1, ..., G 1 5 / B 0 0, B 0 1, ..., B 1 5 は、圧縮伸長プロセスを実行する際の基本単位を構成する。従って、圧縮伸長動作は、コンポーネント R、G、B ( 1 1 1 ) 毎、そしてタイル 1 1 2 毎に、独立して行なわれる。

40

## 【 0 0 2 1 】

ここで、画像データの符号化時、各タイル 1 1 2 のデータは、図 1 に示す色空間変換・逆変換部 1 0 1 に入力され、色空間変換を施された後、2 次元ウェーブレット変換・逆変換部 1 0 2 で 2 次元ウェーブレット変換 ( 順変換 ) が適用されて周波数帯に空間分割される。

50

## 【 0 0 2 2 】

図 3 は、デコンポジションレベル数が 3 であるの場合の各デコンポジションレベルにおけるサブバンドを示す模式図である。2 次元ウェーブレット変換・逆変換部 1 0 2 は、原画像のタイル分割によって得られたタイル原画像 ( 0 L L ) ( デコンポジションレベル 0 ) に対して、2 次元ウェーブレット変換を施し、デコンポジションレベル 1 に示すサブバンド ( 1 L L , 1 H L , 1 L H , 1 H H ) を分離する。そして、2 次元ウェーブレット変換・逆変換部 1 0 2 は、引き続き、この階層における低周波成分 1 L L に対して、2 次元ウェーブレット変換を施し、デコンポジションレベル 2 に示すサブバンド ( 2 L L , 2 H L , 2 L H , 2 H H ) を分離する。2 次元ウェーブレット変換・逆変換部 1 0 2 は、順次同様に、低周波成分 2 L L に対しても、2 次元ウェーブレット変換を施し、デコンポジ  
10

## 【 0 0 2 3 】

次いで、量子化・逆量子化部 1 0 3 では、指定した符号化の順番で符号化の対象となるビットが定められた後、対象ビット周辺のビットからコンテキストが生成される。

## 【 0 0 2 4 】

図 4 は、プレシントを例示する模式図である。量子化の処理が終わったウェーブレット  
20

## 【 0 0 2 5 】

図 5 は、2 次元ウェーブレット変換後の 2 次元ウェーブレット係数の値を「ビットプレーン」単位に分解し、画素或いはコードブロック毎に「ビットプレーン」に順位付けを行う処理の概要を示す模式図である。ウェーブレット変換後の係数値は、そのまま量子化し  
30

## 【 0 0 2 6 】

また、図 5 には、タイル 0 / プレシント 3 / コードブロック 3 について、代表的な「  
40

## 【 0 0 2 7 】

次いで、エントロピー符号化・復号化部 1 0 4 について図 6 を参照しながら説明する。  
図 6 は、符号化された画像データのコードストリームを例示する模式図である。エントロ  
50

ピー符号化・復号化部 104 (図 1 参照) では、コンテキストと対象ビットとから、確率推定によって各コンポーネント RGB のタイル 112 に対する符号化を行う。こうして、原画像の全てのコンポーネント RGB について、タイル 112 単位で符号化処理が行われる。

#### 【0028】

次いで、タグ処理部 105 について説明する。タグ処理部 105 は、エントロピー符号化・復号化部 104 からの全符号化データを 1 本のコードストリームに結合するとともに、それにタグを付加する処理を行う。図 6 に、コードストリームの構造を簡単に示している。このようなコードストリームの先頭と各タイル 112 を構成する部分タイルの先頭には、ヘッダと呼ばれるタグ情報が付加され、その後、各タイル 112 の符号化データが続く。そして、コードストリームの終端には、再びタグが置かれる。

10

#### 【0029】

一方、復号化時には、符号化時とは逆に、各コンポーネント RGB の各タイル 112 のコードストリームから画像データを生成する。このような処理について、図 1 を用いて簡単に説明する。タグ処理部 105 は、外部より入力したコードストリームに付加されたタグ情報を解釈し、コードストリームを各コンポーネント RGB の各タイル 112 のコードストリームに分解し、その各コンポーネント RGB の各タイル 112 のコードストリーム毎に復号化処理を行う。この際、コードストリーム内のタグ情報に基づく順番で復号化の対象となるビットの位置が定められるとともに、量子化・逆量子化部 103 において、その対象ビット位置の周辺ビット (既に復号化を終えている) の並びからコンテキストを生成する。そして、エントロピー符号化・復号化部 104 では、そのコンテキストとコードストリームとから確率推定によって復号化を行なって対象ビットを生成し、それを対象ビットの位置に書き込む。このようにして復号化されたデータは、周波数帯域毎に空間分割されているため、これを 2 次元ウェーブレット変換・逆変換部 102 で 2 次元ウェーブレット逆変換を行うことにより、画像データ中の各コンポーネント RGB における各タイル 112 が復元される。復元されたデータは、色空間変換・逆変換部 101 によって元の表色系のデータに変換される。

20

#### 【0030】

次に、JPEG 2000 の符号フォーマット例を説明する。図 7 は JPEG 2000 の符号フォーマットを示す概略図である。当該符号フォーマットは、符号データの始まりを示す SOC (Start of Codestream) マーカで始まり、その後、符号化のパラメータや量子化のパラメータを記述したメインヘッダが続き、さらに、実際の符号データが続く構成である。実際の符号データは、SOT (Start of Tile-part) マーカで始まり、タイルヘッダ、SOD (Start of Data) マーカ、タイルデータ (符号) で構成される。これら画像全体に相当する符号データの後に、符号の終了を示す EOC (End of Codestream) マーカが付加される。

30

#### 【0031】

図 8 に、メインヘッダの構成例を示す。メインヘッダは、COD, QCD なる必須マーカセグメントと、COC, QCC, RGN, POC, PPM, TLM, PLM, CRG, COM なるオプションマーカセグメントとにより構成されている。

40

#### 【0032】

また、図 9 にタイルヘッダの構成例を示す。図 9 (a) は、タイルヘッダの先頭に付加されるマーカセグメント列を示し、COD, COC, QCD, QCC, RGN, POC, PPT, PLT, COM のマーカセグメント (全てオプション) が使用可能である。一方、図 9 (b) は、タイル内が複数に分割されている場合における分割されたタイル部分列の先頭に付加されるマーカセグメント列であり、POC, PPT, PLT, COM のマーカセグメント (全てオプション) が使用可能である。

#### 【0033】

ここで、JPEG 2000 で使用されるマーカ及びマーカセグメントについて説明する。マーカは 2 バイト (先頭バイトが 0xff で、続くバイトが 0x01 ~ 0xfe) で構

50

成される。マーカ及びマーカセグメントは、以下に示す 6 種類に分類できる。

【 0 0 3 4 】

- ( 1 ) フレーム区切り ( delimiting )
- ( 2 ) 画像の位置、サイズ関係の情報 ( fixed information )
- ( 3 ) 符号化機能の情報 ( functional )
- ( 4 ) エラー耐性用 ( in bit stream )
- ( 5 ) ビットストリームのポインタ ( pointer )
- ( 6 ) 補助的な情報 ( informational )

このうち、本発明に係るマーカは ( 1 ) ( 2 ) である。その詳細について、以下に説明する。

10

【 0 0 3 5 】

まず、Delimitingマーカ及びマーカセグメントについて説明する。Delimitingマーカ及びマーカセグメントは必須であり、SOC, SOT, SOD, EOCがある。符号開始マーカ ( SOC ) は符号列の先頭に付加される。タイル開始マーカ ( SOT ) は、タイル符号列の先頭に付加される。このSOTマーカセグメントの構成を図 10 に示す。当該マーカセグメントの大きさが記述される Lsot、タイル番号 ( 0 から始まるラスタ順につけられた番号 ) が記述される Isot、タイル長さが記述される Psot、タイル部分番号が記述される TPsot、タイル部分数が記述される TNsotなる内容からなる。

【 0 0 3 6 】

次に、Fixed informationマーカセグメントについて説明する。これは、画像についての情報を記述するマーカで、SIZマーカが該当する。SIZマーカセグメントはSOCマーカの直後に付加される。マーカセグメント長はコンポーネント数に依存する。このSIZマーカセグメントの構成を図 11 に示す。当該マーカセグメントの大きさが記述される Lsiz、符号列の互換性 ( 0 固定、0 以外は予約 ) が記述される Rsiz、reference gridの水平方向サイズが記述される Xsiz、reference gridの垂直方向サイズが記述される Ysiz、reference grid原点からの画像の水平方向オフセット位置が記述される XOsiz、reference grid原点からの画像の垂直方向オフセット位置が記述される YOsiz、タイルの水平方向サイズが記述される XTsiz、タイルの垂直方向サイズが記述される YTsiz、reference grid原点からのタイルの水平方向オフセット位置が記述される XTOsiz、reference grid原点からのタイルの垂直方向オフセット位置が記述される YTOsiz、コンポーネント数が記述される Csiz、( i ) 番目のコンポーネントにおけるビット数と符号ビット数が記述される Ssiz(i)、( i ) 番目のコンポーネントにおける水平方向サンプル数が記述される XRsiz(i)、( i ) 番目のコンポーネントにおける垂直方向サンプル数が記述される YRsiz(i)なる内容により構成される。

20

30

【 0 0 3 7 】

ここで、JPEG 2000における画像エリア、タイルの位置関係について説明する。JPEG 2000では、画像やタイルの位置は、図 11 に示したセグメント構成からも判るように、reference gridと称する座標軸を用いて表現される。画像は図 12 に示すように、reference grid上に、reference gridの左上を原点 ( 0 , 0 ) として、原点と画像の左上との相対位置 ( XOsiz , YOsiz ) で指定される。実際の画像エリアの大きさは、( Xsiz - XOsiz ) × ( Ysiz - YOsiz ) で求められる。

40

【 0 0 3 8 】

また、reference grid上に配置された画像は、符号化の際に、前述したように「タイル」と称する矩形領域に分割されて処理される。reference gridとタイルとの位置関係を図 13 に示す。タイルは単独で符号化、復号化できる必要があるため、タイル境界を越えての画素参照はできない。タイルの位置は、reference gridの原点と最初のタイルの左上との相対位置 ( XT Osiz , YT Osiz ) で指定される。画像オフセット位置とタイルオフセット位置、タイルサイズは

$$\begin{array}{l} 0 \quad XT Osiz \quad X Osiz \\ 0 \quad YT Osiz \quad Y Osiz \end{array}$$

50

$$X T s i z + X T O s i z > X O s i z$$

$$Y T s i z + Y T O s i z > Y O s i z$$

なる関係がある。

【0039】

また、全体のタイル数は、

$$\text{水平方向のタイル数} = (X s i z - X T O s i z) / X T s i z$$

$$\text{垂直方向のタイル数} = (Y s i z - Y T O s i z) / Y T s i z$$

なる関係式で示される。

【0040】

また、「画像」「タイル」「サブバンド」「プリシント」「コードブロック」の関係と、「パケット」と「レイヤ」の関係について図14を参照して整理する。 10

【0041】

まず、物理的な大きさの序列は、

画像 > タイル > サブバンド > プリシント > コードブロック

である。

【0042】

「タイル」とは、画像を矩形に分割したものであり、分割数 = 1 の場合、画像 = タイルとなる。「プリシント」とは、サブバンドを矩形に分割したものをHL, LH, HHの3つのサブバンドについて集めたもの、プリシントは3つで1まとまりとなるが、ただし、LLサブバンドを分割したプリシントは1つで1まとまり)で、大まかには画像中の場所(Position)を表すものである。プリシントはサブバンドと同じサイズにできる。プリシントをさらに矩形に分割したものが「コードブロック」である。 20

【0043】

また、プリシントに含まれる全てのコードブロックから、符号の一部を取り出して集めたもの(例えば、全てのコードブロックの最上位ビットMSBから3枚目までのビットプレーンの符号を集めたもの)が「パケット」である。ここに、上述の“一部”は“空(から)”でもいいので、パケットの中身が符号的には“空(から)”ということもある。全てのプリシント(=全てコードブロック=全てのサブバンド)のパケットを集めると、画像全域の符号の一部(例えば、画像全域のウェーブレット係数の、MSBから3枚目までのビットプレーンの符号)ができるが、これが「レイヤ」である。レイヤは、大まかには画像全体のビットプレーンの符号の一部であるから、復号されるレイヤ数が増えれば画質は上がる。レイヤはいわば画質の単位である。 30

【0044】

従って、全てのレイヤを集めると、画像全域の全てのビットプレーンの符号になる。

【0045】

[実施の形態1について]

次に、本発明を実施するための最良の一形態について説明する。

【0046】

図15は、画像処理装置91の電気的な接続を示すブロック図である。この画像処理装置91には、パーソナルコンピュータ、ワークステーションなどが用いられる。図1に示すように、画像処理装置91は、各種演算を行ない、各部を集中的に制御するCPU92と、各種のROMやRAMからなるメモリ93とが、バス94で接続されている。 40

【0047】

バス94には、所定のインターフェイスを介して、ハードディスクなどの磁気記憶装置95と、マウスやキーボードなどで構成される入力装置96と、LCDやCRTなどの表示装置97と、光ディスクなどの記憶媒体98を読取る記憶媒体読取装置99とが接続され、また、ネットワーク100と通信を行なう所定の通信インターフェイス101が接続されている。なお、通信インターフェイス101は、ネットワーク100を介してインターネットなどのWANに接続可能である。記憶媒体98としては、CDやDVDなどの光ディスク、光磁気ディスク、フレキシブルディスクなどの各種方式のメディアを用いるこ 50

とができる。また、記憶媒体読取装置 99 は、具体的には記憶媒体 98 の種類に応じて光ディスクドライブ、光磁気ディスクドライブ、フレキシブルディスクドライブなどが用いられる。

#### 【0048】

磁気記憶装置 95 には、この発明のプログラムを実施する画像処理プログラムが記憶されている。この画像処理プログラムは、この発明の記憶媒体を実施する記憶媒体 98 から記憶媒体読取装置 99 により読取るか、あるいは、インターネットなどの WAN からダウンロードするなどして、磁気記憶装置 95 にインストールしたものである。このインストールにより画像処理装置 91 は動作可能な状態となる。なお、この画像処理プログラムは、所定の OS 上で動作するものであってもよい。また、特定のアプリケーションソフトの一部をなすものであってもよい。

10

#### 【0049】

このようなハードウェア構成の画像処理装置 91 においては、画像処理プログラムにもとづいて、図 16 ~ 図 21 を参照して後述するような処理を行うことができる。

#### 【0050】

図 16 は、画像処理プログラムにもとづいて実行する処理の機能ブロック図である。この処理は、元画像の指定された位置に対して編集処理を行うものである。画像処理プログラムにより、入力されるブロック単位の符号列（画像データから例えば JPEG2000 アルゴリズムにより圧縮符号化された符号列）から編集対象となる符号列を検索するブロック単位検索手段 31 と、ブロック単位検索手段 31 で抽出された編集対象のブロック単位の符号に対して編集処理を行い、新たな符号列を作成するブロック単位編集処理手段 32 と、ブロック単位編集処理手段 32 で作成された符号列と編集対象ではないブロック単位の符号とから全体の符号列を作成する符号列制御手段 33 が実現される。

20

#### 【0051】

その動作を簡単に説明すると、符号はまずブロック単位検索手段 31 に入力され、編集対象となる画像を含むブロック単位の符号であるか否かを判断する。タイルを使用した JPEG2000 符号を例に取ると、図 17 に示すように、タイル単位の符号列 41 には、先頭にヘッダ情報 42 が存在する。ヘッダ情報 42 の中にはヘッダ番号 43 とその符号列の長さを示す情報 44 が存在する。検索は、このヘッダ情報 42 のみ解釈していけば簡単に行える。例えば、編集対象でなかったら、符号列 41 の長さの情報を利用して次のヘッダ情報 42 を検出していけばよいので、検索は高速に行える。

30

#### 【0052】

ブロック単位検索手段 31 で編集対象と判断されたブロック単位の符号は、ブロック単位編集処理手段 32 に送られる。ブロック単位編集処理手段 32 では、入力された符号に対して、指定された所定の編集処理を実施する。編集処理の流れとしては、図 18 に示すように、まず、該当符号を復号してブロック単位の符号を復元し（ステップ S1）、続いて、復元された画像に対して編集処理を行う（ステップ S2）。最後に、編集後の画像に対して元の符号と同じ方法で符号化を行い、元の符号列と同じ形式の符号列を作成する（ステップ S3）。編集処理の一例としては、色の変更や、画像の合成等が挙げられる。この編集処理後の新たに作成された符号列は、符号列制御手段 33 に送られる。符号列制御手段 33 では、編集対象でない元のブロック単位の符号列と、ブロック単位編集処理手段 32 で新たに作成されたブロック単位の符号列とを結合させ、全体としてフォーマット整合された符号列を作成する。

40

#### 【0053】

この場合の符号列の結合方法については様々な方法が考えられるが、符号量に応じて適切な手段適宜選択するのが望ましい。図 19 は、その一例を説明する符号列制御手段 33 のブロック図である。この例では、編集後の符号量が編集前の符号量より多い場合には結合方式 A を実行する結合手段 52 を、少ない場合には結合方式 B を実行する結合手段 53 を選択して、符号の結合を行う。すなわち、編集の前後の符号量の比較を符号量比較手段 51 で行い、結合手段 52 と 53 とをセレクタ 54 で選択する。なお、結合処理の影響を

50

受けないブロック単位の符号は何も処理を行なわない。

【0054】

上記のような処理を行なえば、符号列中の特定部分の符号のみ処理することができる。また、編集後の新たな符号列の挿入方法を、編集前の符号を参照しながら、最短処理時間で行えるように全体符号列を制御することができるので、高速な編集処理が実現することができることになる。

【0055】

符号列制御手段33における、編集対象でない元のブロック単位の符号列と、編集後の新たに作成されたブロック単位の符号列とを結合させる符号列結合の具体的な手段については様々な方法が考えられる。

【0056】

まず、新たに作成された符号列の符号量が、編集前の符号量を超えてしまった場合には、符号列の最後に新たに作成された符号列を追加する形で、全体の符号列を作成する方法が挙げられる。一例として、図20に示すように、編集前(a)の符号部分には、その部分が符号列の最後に追加されていることを表す識別情報を入れると共に、追加された符号列の先頭アドレスを入れておく。こうすれば、追加された符号列(後に符号列全体の再配列を考慮して、編集後(b)のブロック単位の画像に対応する正規の符号列とする)に対して高速にアクセスが可能となる。上記の処理を行っても、元のヘッダ情報の書き換えを行っていないので、符号フォーマットとしては問題がない。この方法によれば、元の位置に符号を挿入するために、それ以降のデータを再配置しなければならないという処理上の問題はなくなり、高速な編集処理が実現できる。

【0057】

なお、図20の例では、新たな符号列を、終了符号の直後に配置したが、図21に示すように、終了符号の直前に挿入しても良い。ただし、その場合には、最終のブロック単位の符号列における符号長の情報を操作(最終の符号列の符号長に新たな符号列の符号長を足す)して、全体の符号列としてフォーマット整合させる必要がある。

【0058】

また、結合する際に問題とならないように、編集後のブロックの符号量が多くなる場合には、符号化パラメータを変更して編集前の符号量を超えないような符号列を作成することが考えられる。特に、JPEG2000方式は、同じ構成でlosslessおよびlossyが可能なスケーラブルな符号化方式であり、符号のまま符号量の調節が可能である。この性質を利用すれば、一度作成した符号列に対して、符号のまま符号量変更が可能であるため、編集前の符号量に最も近く、かつ、最高となる画質を得られる符号列を高速に作成することができる。この際、編集後の符号量が編集前の符号量に一致しない場合でも、編集後のヘッダ情報42は、編集前の符号長と同じにしておく。こうすれば、符号全体として問題はない。

【0059】

以上のような手段により、符号列の編集処理を高速に行い、かつ、編集後の符号列を高速に作成することは可能になるが、符号フォーマットの観点から言えば、本来あるべき姿に対して冗長な部分が存在していることになる。例えば、図20、図21を参照して説明した編集後のブロックの符号量が少ない場合の処理で作成される符号では、編集後の符号列は末尾に追加する形で存在しているし、前述の編集後のブロックの符号量が多い場合の処理で作成される符号列では、編集前の符号列に満たない部分の符号は意味のない符号データとなっている。

【0060】

この点を考慮すると、全体の処理時間を考慮して、必要であれば適宜、必要な部分のみの符号列を再配置することで、本来あるべき姿の符号列にするのが望ましい。具体的には、符号列制御手段33によって、全体の処理時間を考慮しながら編集後の符号列の挿入方法を決定し、必要であれば、適宜、必要部分のみ符号列を再配置することが考えられる。この方法によれば、処理速度を低下させることなく、無駄な符号情報を削除することが可

10

20

30

40

50

能となり、効率の良い符号列を作成可能となる。

【0061】

[実施の形態2について]

本発明の別の実施の形態を本発明の実施の形態2として説明する。

【0062】

図22は、本発明の画像処理装置を実施した複写機1の電氣的な接続を示すブロック図である。原稿を光学的に読み取るスキャナである読み取りユニット11は、原稿に対するランプ照射の反射光をミラーおよびレンズなどからなる光学系によりCCD(電荷結合素子)などの光電変換素子に集光する。この光電変換素子は、SBU(センサ・ボード・ユニット)12に搭載され、受光素子において電気信号に変換された画像信号はデジタル画像信号に変換された後、SBU12から出力される。SBU12から出力される画像信号はCDIC13(圧縮/伸長およびデータインターフェイス制御部)13に入力される。機能デバイスおよびデータバス間における画像データの伝送はCDIC13が全て制御する。CDIC13は画像データに関し、SBU12、パラレルバス14、IPP(画像処理プロセッサ)15間のデータ転送、本システムの全体制御を司るシステムコントローラ(CPU)16と画像データに対するプロセスコントローラ27間の通信を行なう。符号16a, 16bは、システムコントローラ16が使用するROM、RAMである。それぞれSBU12からの画像信号は、CDIC13を経由してIPP15に転送され、光学系およびデジタル画像信号への量子化に伴う信号劣化(スキャナ系の信号劣化とする)が補正されて、再度CDIC13へ出力される。

10

20

【0063】

この複写機1では、読み取りユニット11による読み取り画像をメモリに蓄積して再利用するジョブと、メモリに蓄積しないジョブとがあり、以下ではそれぞれの場合について説明する。メモリに蓄積する例としては、1枚の同一原稿を複数枚複写する場合、読み取りユニット11で1回だけ原稿の読取動作を行い、メモリに蓄積し、蓄積データを複数回読み出す使い方がある。メモリを使わない例としては、1枚の原稿を1枚だけ複写する場合、読み取り画像をそのまま印刷すればよいので、メモリアクセスを行なう必要はない。

【0064】

まず、メモリを使わない場合、IPP15からCDIC13へ転送された画像データは、再度CDIC13からIPP15へ戻される。IPP15において受光素子による輝度データを面積階調に変換するための画質処理を行なう。この画質処理後の画像データはIPP15からVDC(ビデオ・データ制御)17に転送する。そして、面積階調に変化された信号に対し、ドット配置に関する後処理およびドットを再現するためのパルス制御を行い、電子写真方式で画像形成するプリンタエンジンである作像ユニット18により、転写紙上に再生画像を形成する。なお、作像ユニット18の印刷方式は、電子写真方式のほか、インクジェット方式、昇華型熱転写方式、銀塩写真方式、直接感熱記録方式、熔融型熱転写方式など、様々な方式を用いることができる。

30

【0065】

メモリに画像データを蓄積し、画像データの読み出し時に付加的な処理、例えば、画像方向の回転、画像の合成等を行なう場合の画像データの流れを説明する。IPP15からCDIC13へ転送された画像データは、CDIC13からパラレルバス14を経由してIMAC(画像メモリアクセス制御)19に送られる。IMAC19では、システムコントローラ16の制御に基づき、画像データの記憶装置であるMEM(メモリモジュール)20へのアクセス制御、外部のPC(パソコン)21へのプリント用データの展開、MEM20のメモリ有効活用のための画像データの圧縮/伸長を行なう。IMAC19へ送られた画像データはデータ圧縮後MEM20へ蓄積され、この蓄積データは必要に応じて読み出される。読み出した画像データは伸長されて本来の画像データに戻され、IMAC19からパラレルバス経由でCDIC13へ戻される。

40

【0066】

CDIC13からIPP15への転送後は画像データに対して画質処理およびVDC1

50

7でのパルス制御を行い、その画像データにより作像ユニット18において転写紙上に画像形成する。

【0067】

この複写機1は、いわゆる複合機であり、FAX送信機能を備えている。このFAX送信機能は、読み取り画像データにIPP15にて画像処理を実施し、CDIC13およびパラレルバス14を経由してFCU(FAX制御ユニット)22へ転送する。FCU22にて通信網へのデータ変換を行い、PN(公衆回線)23へFAXデータとして送信する。FAX受信は、PN23からの回線データをFCU22で画像データへ変換し、パラレルバス14およびCDIC13を経由してIPP15へ転送する。この場合、特別な画質処理は行なわず、VDC17においてドット再配置およびパルス制御を行い、作像ユニット18において転写紙上に再生画像を形成する。

10

【0068】

複数のジョブ、例えば、コピー機能、FAX送受信機能、プリンタ出力機能が並行に動作する状況において、読み取りユニット、作像ユニットおよびパラレルバス14の使用権のジョブへの割り振りをシステムコントローラ16およびプロセスコントローラ27で制御する。

【0069】

プロセスコントローラ(CPU)27は画像データの流れを制御し、システムコントローラ16はシステム全体を制御し、各リソースの起動を管理する。符号27a, 27bは、プロセスコントローラ27が使用するROM、RAMである。

20

【0070】

ユーザは、操作パネル24を選択入力することで各種の機能の選択を行ない、コピー機能、FAX機能等の処理内容を設定する。

【0071】

システムコントローラ16とプロセスコントローラ27はパラレルバス14、CDIC13およびシリアルバス25を介して相互に通信を行なう。この際、CDIC13内においては、パラレルバス14とシリアルバス25とのデータインターフェイスのためのデータフォーマット変換を行なう。

【0072】

MLC(Media Link Controller)26は、画像データの符号変換の機能を実現する。具体的にはCDIC13で使用される符号化方式、IMAC19で使用される符号化方式から他の符号化方式への変換(例えば、標準であるJPEG方式等)を行なう。

30

【0073】

この複写機1では、実施の形態1において、図16~図21を参照して前述した符号列の編集処理を、操作パネル24の操作により、システムコントローラ16が画像処理プロセッサ15を制御して画像処理プロセッサ15で実行する。よって、その処理内容は実施の形態1と同様であるため、詳細な説明は省略する。

【図面の簡単な説明】

【0074】

【図1】本発明の実施の形態の前提となるJPEG2000方式の基本となるアルゴリズムを実現するシステムの機能ブロック図である。

40

【図2】原画像の各コンポーネントの分割された矩形領域を示す説明図である。

【図3】デコンポジションレベル数が3の場合の、各デコンポジションレベルにおけるサブバンドを示す説明図である。

【図4】プレシントを示す説明図である。

【図5】ビットプレーンに順位付けする手順の一例を示す説明図である。

【図6】符号化された画像データのコードストリームを例示する模式図である。

【図7】JPEG2000の符号フォーマットを示す概略図である。

【図8】そのメインヘッダの構成図である。

【図9】タイルヘッダの構成図である。

50

- 【図10】SOTマーカセグメントの構成図である。
- 【図11】SIZマーカセグメントの構成図である。
- 【図12】画像の位置関係を示す説明図である。
- 【図13】Reference gridとタイルとの位置関係を示す説明図である。
- 【図14】画像、タイル、サブバンド、プリシнкт、コードブロックの関係を示す説明図である。
- 【図15】本発明の実施の形態1の画像処理装置の電気的な接続のブロック図である。
- 【図16】符号列を編集処理する手段の機能ブロック図である。
- 【図17】符号列の構成について説明する説明図である。
- 【図18】符号列を編集処理する手順のフローチャートである。
- 【図19】符号列制御手段の機能ブロック図である。
- 【図20】新たに作成された符号列の符号量が、編集前の符号量を超えてしまった場合には、符号列の最後に新たに作成された符号列を追加する形で、全体の符号列を作成する方法のひとつの説明図である。
- 【図21】図20の場合の他の例の説明図である。
- 【図22】本発明の実施の形態2の複合機の電気的な接続のブロック図である。

10

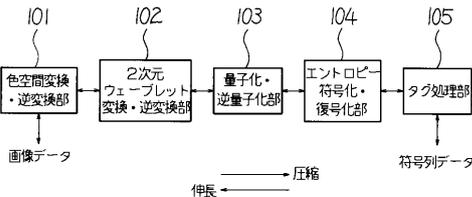
20

【符号の説明】

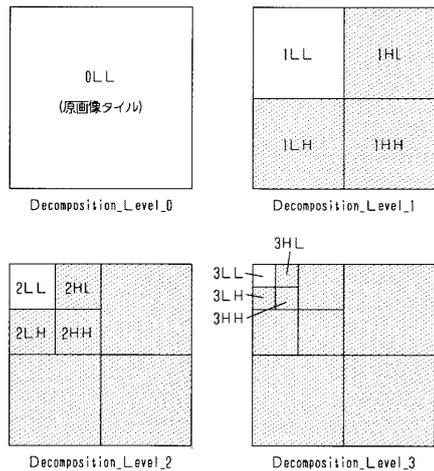
【0075】

- 1 画像処理装置
- 91 画像処理装置
- 98 記憶媒体

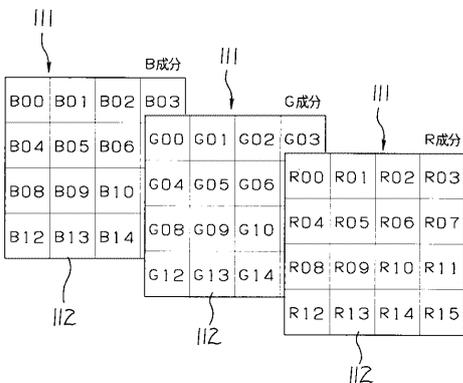
【図1】



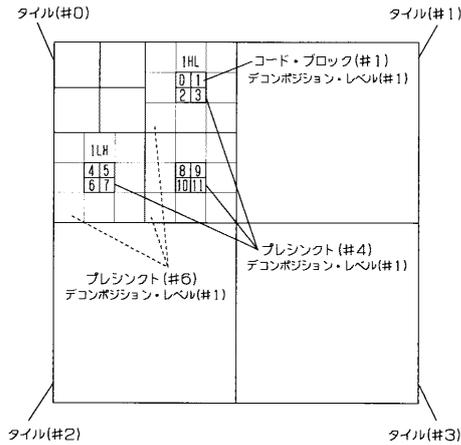
【図3】



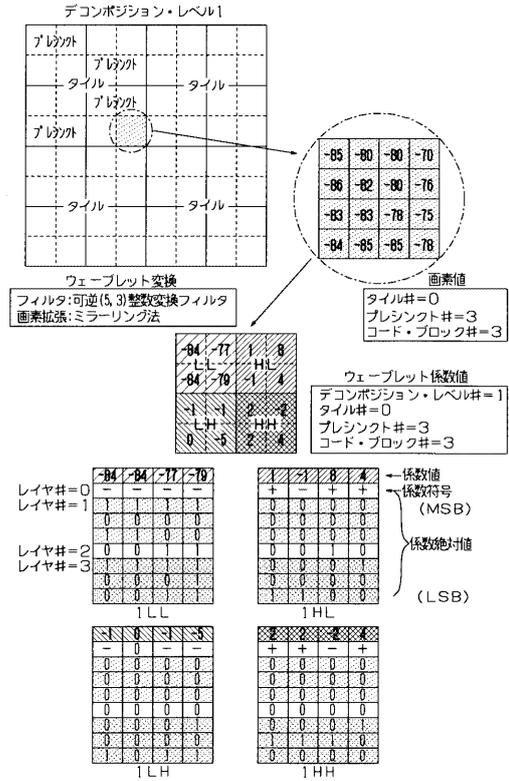
【図2】



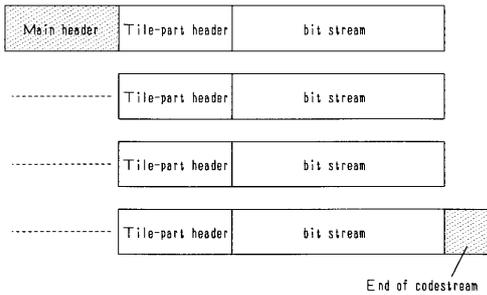
【 図 4 】



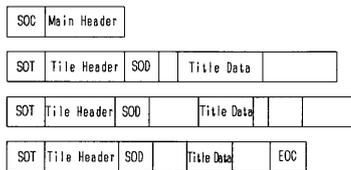
【 図 5 】



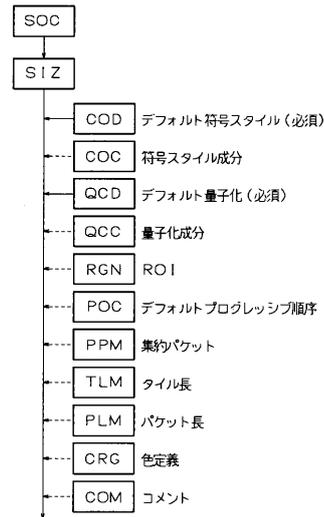
【 図 6 】



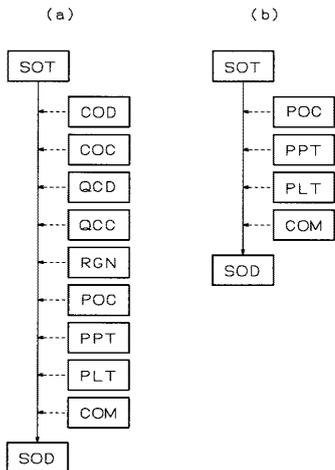
【 図 7 】



【 図 8 】



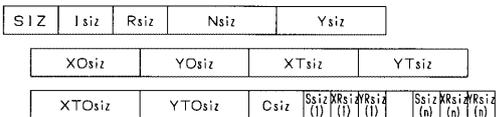
【図9】



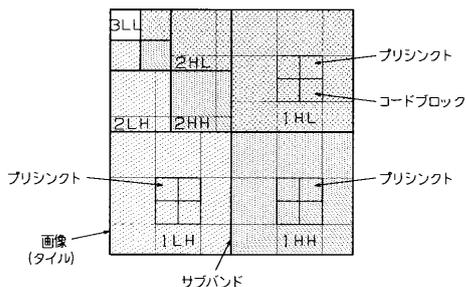
【図10】



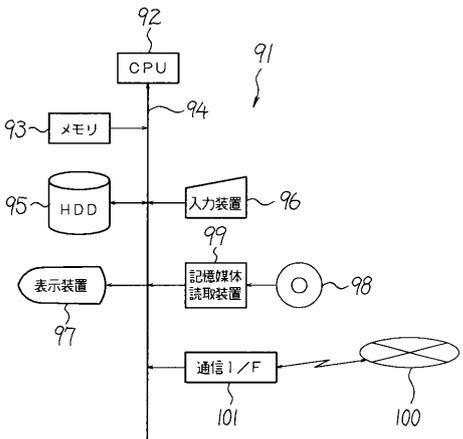
【図11】



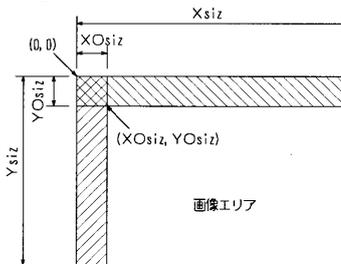
【図14】



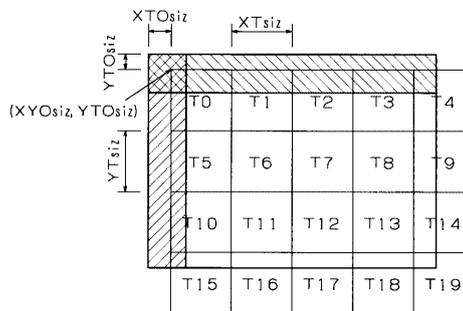
【図15】



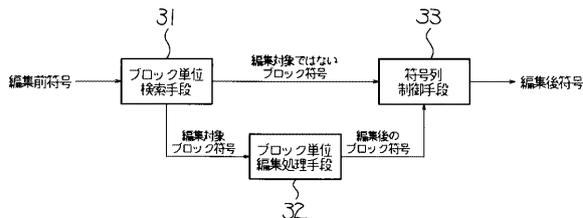
【図12】



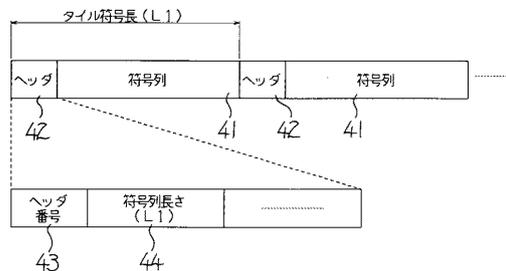
【図13】



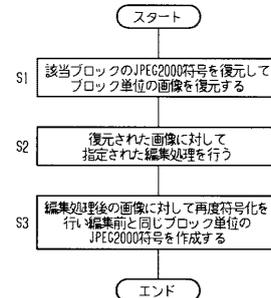
【図16】



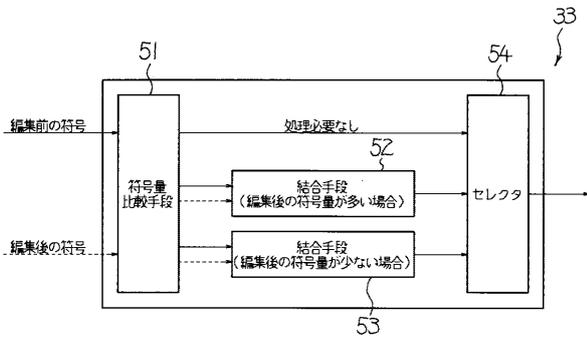
【図17】



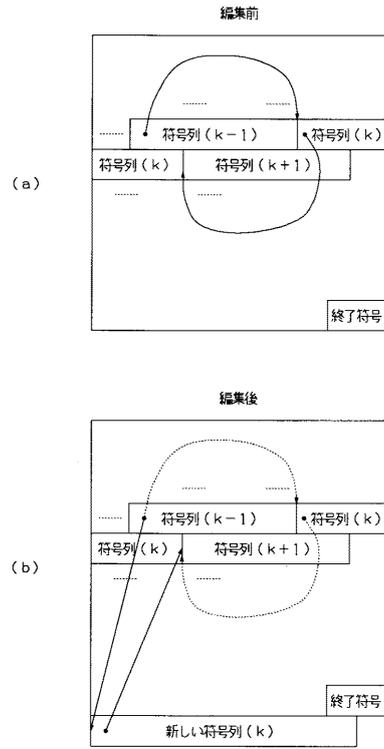
【図18】



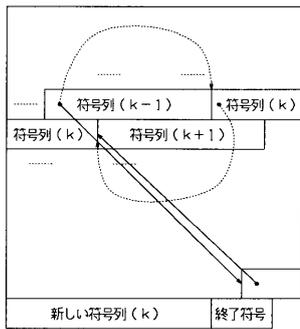
【 図 1 9 】



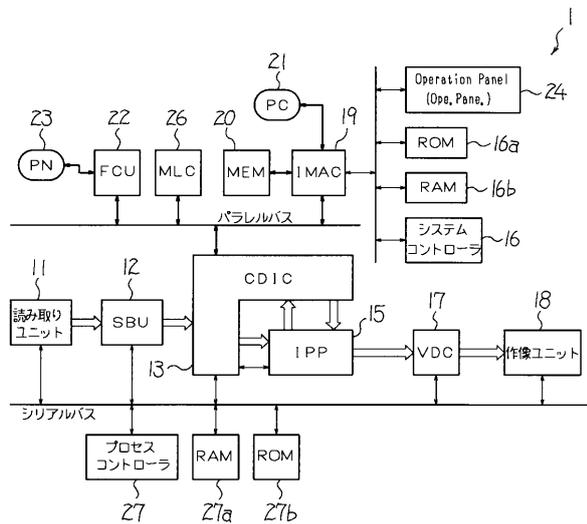
【 図 2 0 】



【 図 2 1 】



【 図 2 2 】



フロントページの続き

Fターム(参考) 5C078 AA09 BA44 BA53 CA01 CA14 CA25 CA31 DA01 DA06 DA07