

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6635751号  
(P6635751)

(45) 発行日 令和2年1月29日(2020.1.29)

(24) 登録日 令和1年12月27日(2019.12.27)

(51) Int.Cl.	F 1
HO4N 19/51 (2014.01)	HO4N 19/51
HO4N 19/593 (2014.01)	HO4N 19/593
GO6T 1/00 (2006.01)	GO6T 1/00 290B

請求項の数 9 (全 20 頁)

(21) 出願番号	特願2015-210005 (P2015-210005)
(22) 出願日	平成27年10月26日 (2015.10.26)
(65) 公開番号	特開2017-85285 (P2017-85285A)
(43) 公開日	平成29年5月18日 (2017.5.18)
審査請求日	平成30年10月22日 (2018.10.22)

(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(74) 代理人	100076428 弁理士 大塚 康徳
(74) 代理人	100115071 弁理士 大塚 康弘
(74) 代理人	100112508 弁理士 高柳 司郎
(74) 代理人	100116894 弁理士 木村 秀二
(74) 代理人	100130409 弁理士 下山 治
(74) 代理人	100134175 弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 画像符号化装置及びその制御方法、並びにプログラム及び記憶媒体

## (57) 【特許請求の範囲】

## 【請求項 1】

複数の画像データで構成される画像データセットを符号化する画像符号化装置であって、

前記複数の画像データにおける符号化対象の着目画像データから、 $M \times N$  画素で構成される符号化対象の着目ブロックを入力し、当該着目ブロックに近似する近似ブロックを既符号化画像データ内から探し、探索した近似ブロックに対する前記着目ブロックの相対位置をベクトルとして抽出する抽出手段と、

前記着目ブロック外の符号化済みの画素を含む拡張領域と前記着目ブロックで構成される  $(M + P) \times (N + Q)$  画素の拡張ブロックと、前記近似ブロックを包含する  $(M + P) \times (N + Q)$  画素の参照拡張ブロックとから、差分拡張ブロックを生成する生成手段と、

該差分拡張ブロックにおける前記拡張領域内の差分画素データを参照してコンテキストを生成し、前記差分拡張ブロック内の前記着目ブロックと前記近似ブロックとの差分である差分ブロックを構成する各差分画素データを、生成した前記コンテキストに基づき予測符号化する符号化手段と

を有することを特徴とする画像符号化装置。

## 【請求項 2】

符号化手段は、前記差分ブロックを構成する各差分画素データを可逆符号化することを特徴とする請求項 1 に記載の画像符号化装置。

**【請求項 3】**

前記 P、Q は 1 であって、前記符号化手段は J P E G - L S に準拠して予測符号化することを特徴とする請求項 1 または 2 に記載の画像符号化装置。

**【請求項 4】**

前記符号化手段は、前記拡張領域の差分画素データに基づき、予め用意された複数のコンテキストセットの中から前記着目ブロックで利用するコンテキストセットを選択し、予測符号化を行うことを特徴とする請求項 1 に記載の画像符号化装置。

**【請求項 5】**

前記符号化手段は、

前記着目ブロックに隣接し既符号化の複数の隣接ブロックを符号化した際に求めたベクトルそれぞれと、前記着目ブロックのベクトルとの比較する比較手段と、

該比較手段の比較結果に基づき、前記着目ブロックに対応する差分ブロックを、当該差分ブロックの前記拡張領域内のコンテキストを利用して予測符号化するか、前記隣接ブロックを符号化した際に利用したコンテキストを利用した予測符号化するかを切り換える手段と

を含むことを特徴とする請求項 1 に記載の画像符号化装置。

**【請求項 6】**

前記画像データセットは、C T または M R I によって得られた被検体の体軸に沿った複数の断層画像データであることを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の画像符号化装置。

10

20

**【請求項 7】**

コンピュータが読み込み実行することで、前記コンピュータに、請求項 1 乃至 6 のいずれか 1 項に記載の画像符号化装置が有する各手段として機能させるためのプログラム。

**【請求項 8】**

請求項 7 に記載のプログラムを格納した、コンピュータが読み込み可能な記憶媒体。

**【請求項 9】**

複数の画像データで構成される画像データセットを符号化する画像符号化装置の制御方法であって、

抽出手段が、前記複数の画像データにおける符号化対象の着目画像データから、 $M \times N$  画素で構成される符号化対象の着目ブロックを入力し、当該着目ブロックに近似する近似ブロックを既符号化画像データ内から探索し、探索した近似ブロックに対する前記着目ブロックの相対位置をベクトルとして抽出する抽出工程と、

生成手段が、前記着目ブロック外の符号化済の画素を含む拡張領域と前記着目ブロックで構成される  $(M + P) \times (N + Q)$  画素の拡張ブロックと、前記近似ブロックを包含する  $(M + P) \times (N + Q)$  画素の参照拡張ブロックとから、差分拡張ブロックを生成する生成工程と、

符号化手段が、該差分拡張ブロックにおける前記拡張領域内の差分画素データを参照してコンテキストを生成し、前記差分拡張ブロック内の前記着目ブロックと前記近似ブロックとの差分である差分ブロックを構成する各差分画素データを、生成した前記コンテキストに基づき予測符号化する符号化工程と

30

40

を有することを特徴とする画像符号化装置の制御方法。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、画像データの符号化技術に関するものである。

**【背景技術】****【0002】**

近年、C T (Computed Tomography) や M R I (Magnetic Resonance Imaging) などの医療機器の撮影技術の発達により、撮影時間の短縮や、スライスの高密度化、撮影条件の多様化が進んでおり、断層画像（以下、スライス画像データとも呼ぶ）の枚数が増大している

50

。人体の胸部や腹部など、ある一つの部位の撮影だけでも、2000枚から3000枚の断層画像が生成されるようなケースも増えてきている。そのため、断層画像の総データ量も膨大になるため、圧縮技術が重要になる。

#### 【0003】

従来、撮影で得られた断層画像の1枚1枚は、JPEGの可逆方式やJPEG2000などの、国際標準の可逆符号化技術を用いて圧縮している。しかしながら、各断層画像を個別に可逆符号化したのでは、圧縮効果はそれほど期待できない。

#### 【0004】

近年の撮影技術の発達により、1mm厚や2mm厚など、断層画像がかなり高密度になっている。そのため、符号化対象の断層画像と、それに隣接する断層画像との差分画像を生成し、生成した差分画像を符号化することで、圧縮性能を更に向上させる技術が提案されている（特許文献1）。

10

#### 【0005】

また、動画符号化の標準技術であるH.264などでは、ブロック単位に動き補償予測を行うことで、フレーム間相関を除去し、符号化効率を向上させる技術を採用している。一般的に、動き補償予測を行ったブロックに対しては周波数変換を行って符号化するなど、ブロック独立で処理がなされる（非特許文献1）。

#### 【先行技術文献】

#### 【特許文献】

#### 【0006】

20

【特許文献1】特開2009-13643号公報

#### 【非特許文献】

#### 【0007】

【非特許文献1】ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0008】

動き補償予測と、着目画素の周囲画素から予測値を生成して予測符号化するような方式を組み合わせて符号化を行う場合には、動き補償予測の処理単位であるブロック間の境界で画素値が不連続となり、予測誤差が大きくなり、その結果、符号化効率が低下する。

30

#### 【0009】

本発明はかかる点に鑑みてなされたものであり、CTやMRIなど、被検体の体軸に沿って得られた複数の画像データで構成される画像データセットを、高い圧縮率で符号化する技術を提供しようとするものである。

#### 【課題を解決するための手段】

#### 【0010】

この課題を解決するため、例えば本発明の画像符号化装置は以下の構成を備える。すなわち、

複数の画像データで構成される画像データセットを符号化する画像符号化装置であって、

40

前記複数の画像データにおける符号化対象の着目画像データから、 $M \times N$ 画素で構成される符号化対象の着目ブロックを入力し、当該着目ブロックに近似する近似ブロックを既符号化画像データ内から探索し、探索した近似ブロックに対する前記着目ブロックの相対位置をベクトルとして抽出する抽出手段と、

前記着目ブロック外の符号化済みの画素を含む拡張領域と前記着目ブロックで構成される $(M + P) \times (N + Q)$ 画素の拡張ブロックと、前記近似ブロックを包含する $(M + P) \times (N + Q)$ 画素の参照拡張ブロックとから、差分拡張ブロックを生成する生成手段と、

該差分拡張ブロックにおける前記拡張領域内の差分画素データを参照してコンテキストを生成し、前記差分拡張ブロック内の前記着目ブロックと前記近似ブロックとの差分であ

50

る差分ブロックを構成する各差分画素データを、生成した前記コンテキストに基づき予測符号化する符号化手段とを有する。

【発明の効果】

【0011】

本発明によれば、CTやMRIなど、被検体の体軸に沿って得られた複数の画像データで構成される画像データセットを、高い圧縮率で符号化することが可能になる。

【図面の簡単な説明】

【0012】

【図1】第1の実施形態に係る画像処理装置の機能構成を示すブロック図。

【図2】符号化対象画像データとブロックの関係を表す図。

10

【図3】符号化対象ブロックと拡張領域の画素の位置関係を示す図。

【図4】差分画像ブロックデータの例と、着目画素Xを予測符号化の際の参照画素との関係を示す図。

【図5】符号化対象ブロックと拡張領域との関係を示す図。

【図6】参照拡張ブロック、拡張ブロック周辺、差分拡張ブロックとの関係を示す図。

【図7】第1の実施形態に係る参照ブロックの特定方法の一例を示す図。

【図8】第1の実施形態に係る参照ブロックの設定エリアについて示す図。

【図9】コンテキスト生成時の量子化閾値の説明図。

【図10】第4の実施形態に係るタイプ分類を示す図。

【図11】第4の実施形態に係る予測誤差のヒストグラムを表す図。

20

【図12】第5の実施形態に係る各ブロックの動きベクトルの例を示す図。

【図13】第1の実施形態の変形例における符号化処理のフローチャート。

【図14】第1の実施形態の変形例の情報処理装置のブロック図。

【図15】第2の実施形態に係る画像処理装置のブロック図。

【図16】本画像処理装置の出力符号列の構成を示す図。

【図17】実施形態における画像データセットの例を示す図。

【発明を実施するための形態】

【0013】

以下添付図面を参照して、本発明を好適な実施形態に従って詳細に説明する。なお、以下の実施形態において示す構成は一例にすぎず、本発明は図示された構成に限定されるものではない。

30

【0014】

【第1の実施形態】

図1は、本第1の実施形態に係る画像処理装置の機能構成を示すブロック図である。図1に示すように、本実施形態に係る画像処理装置は、バッファ101、ブロック分割部102、ブロックバッファ103、動き補償部104、拡張ブロック生成部105、差分画像生成部106、予測符号化部107、符号列形成部108を有する。同図において109、110は信号線を示す。

【0015】

以下、図1を参照して、本実施形態に係る画像処理装置が行う処理について説明する。本第1の実施形態に係る画像処理装置の符号化対象は、複数枚の画像データからなる画像データセットである。画像データセットを構成する画像データは、水平方向W画素、垂直方向H画素、モノクロ多値で構成されているものとする。本実施形態では、画素値は2バイト(16ビット)のデータ形式であり、そのうち12ビットが有効ビットとする。つまり、1画素は4096( $=2^{12}$ )階調の輝度値で表されるものとする。ただし、これらの数値は、あくまで理解を容易にするための具体例であると認識されたい。

40

【0016】

次に、本実施形態の画像処理装置での各部の動作について説明する。符号化対象画像データセットは信号線109を介してバッファ102に一旦格納される。符号化対象画像データの供給源は特に問わない。また、本実施形態における符号化対象画像データは、CT

50

やM R Iなどの医療機器で撮影されたデータ（生データ）を再構成して得られる画像データとする。なお、医療におけるデジタル画像と通信の標準規格としてD I C O M (Digital Imaging and Communication in Medicine) が知られている。D I C O Mでは各画像データセットのスライス間間隔やスライス厚さの情報に関しては、ファイル内のタグデータなどから取得できる。実施形態における符号化対象の画像データセットは、このD I C O Mに準拠したものとして説明する。図17は、バッファ102に格納されD I C O Mのファイル内に含まれる画像データセットの概念図である。画像データセットは、被検体の体軸に沿った複数の画像（スライス画像データ、もしくは断層画像データ）で構成され、画像のスライス間隔、スライス厚さは、ファイル内のタグから得られる。図示の場合、画像データセットは被験者の体軸に沿った1000枚の画像で構成され、個々の画像S(0)、S(1)、…S(999)として特定する。互いに隣り合う画像は、被験者の体軸に対して近接した位置の画像であるので、互いに高い相関があるとみなせる。つまり、本願発明者は、被験者の体軸方向のベクトルが、動画像における時間軸のベクトルと同じであると考えた。それ故、実施形態では、動画像における動き補償符号化の思想を用いて、これら画像データセットを符号化する。

#### 【0017】

ここで先頭の画像S(0)は、その画像を単独に可逆符号化を行う。そして、可逆符号化であれば、特にその種類は問わないが、例えばJ P E G - L Sの可逆モードを利用する。そして、先頭以外の画像S(k) (k = 1, 2, ..., 999のいずれか)を符号化する際には、その直前に符号化した画像S(k-1)を参照画像として、可逆の予測符号化を行う。なお、上記の通りなので、バッファ101は最低でも符号化対象の着目画像と参照する画像（実施形態では直前に符号化した画像）の2つの画像が格納できればよく、画像データセットに含まれる全画像をバッファ101に保持する必要はない。また、先頭画像S(0)を符号化する際には、全画素値が所定値（例えば0”）の仮想的な画像S(-1)を参照画像として符号化しても構わない。

#### 【0018】

ブロック分割部102は、バッファ101へ格納された符号化対象画像データS(k)を複数の矩形ブロックに分割する。1つの矩形ブロックのサイズは水平方向M画素、垂直方向N画素とする。そして、ブロック分割部102は、バッファ101から、分割された矩形ブロック単位に読み出してブロックバッファ103へと格納する。ブロック分割部102がバッファ101からブロックを読み出す順番は、ブロックを単位とするラスタースキャン順とする。説明の便宜上、符号化対象の各画像データS( )の水平方向の画素数WはMの整数倍であり、画像データS( )の垂直方向の画素数HもNの整数倍であるとする。つまり、符号化対象画像データを矩形ブロックに分割した場合に不完全な矩形ブロックが発生しないものとする。また、この水平方向M画素、垂直方向N画素で構成される矩形ブロックを、以降、単にブロックもしくはブロックデータと呼ぶ。

#### 【0019】

図2に符号化対象画像データとブロックの関係を図示する。図のように符号化対象画像データS(k)において、左上隅を原点とし、水平方向i番目、垂直方向j番目のブロックをB(i, j)と記す。そして、ブロックB(i, j)内の水平方向m番目、垂直方向n番目の画素をB<sub>i, j</sub>(m, n)と記す。例えば、ブロックB(2, 3)内の左上隅の画素は、B<sub>2, 3</sub>(0, 0)と表す。

#### 【0020】

ブロック分割部102は、ブロックバッファ103に、符号化対象のブロックデータを順次格納していく。このとき、ブロックバッファ103に格納するデータは、着目ブロックを構成するM×N画素に加えて、着目ブロック外の、当該着目ブロックに接する上1ラインと左1列の画素データも格納する。つまり、ブロックバッファ103に必要とされる最低容量は(M+1)×(N+1)×2（各画素2バイトのため）+2バイトとなる。最後に2バイト（1画素）が追加されている理由は、実施形態ではJ P E G - L Sを用いることを前提にしているからである。J P E G - L Sでは、図4に示すように、符号化対象

10

20

30

40

50

画素を X としたとき、画素 X に隣接する、符号化済みの画素値 a , b , c , d を参照する。この場合の隣接画素 d は、着目画素 X の水平右方向の座標より 1 つ先行した位置にあるためである。

【 0 0 2 1 】

実施形態では説明をより単純化するため、M = N = 8 として説明する。図 3 は、ブロックバッファ 103 に格納されるブロックと、図示の斜線部で示すブロック外の画素データとの関係を示している。以降、この斜線部を拡張領域という。なお、着目ブロック内の左上隅の画素の、さらに左上の拡張領域内の画素は  $B_{ij}(-1, -1)$  と表すものとする。そして、拡張領域と着目ブロックとを合わせたブロックを以降、拡張ブロックと呼ぶ。  $B_{ij}(8, -1)$  は先の説明における最後に追加されている 2 バイト分にあたる。符号化対象ブロック  $B(i, j)$  を符号化する際には、既に符号化済の領域となっているため、これを拡張領域内の画素と同様に扱うことができる。  
10

【 0 0 2 2 】

なお、着目ブロックが符号化対象画像の左端や上端等に位置するとき、拡張領域の画素の一部または全部が実在しないことになる。この場合、その実在しない画素値は所定値であると見なす。この所定値は、符号化装置と復号装置との間で共通であればよいので、特に制限はないが、ここでは“0”であるものとする。

【 0 0 2 3 】

動き補償部 104 は、バッファ 101 に格納された画像データセットの中から、着目画像データ  $S(k)$  の直前に符号化した既符号化画像データ  $S(k-1)$  を読み出す。そして、動き補償部 104 は、ブロックバッファ 103 内の格納された符号化対象のブロック  $B(i, j)$  との差分が最小となるブロックの位置を既符号化画像データ内から探索もしくは特定する。そして、動き補償部 104 は、探索したブロックに対する着目ブロックの相対位置を動きベクトルとして抽出する。なお、既符号化画像データ  $S(k-1)$  は、着目画像データ  $S(k)$  を符号化する際に参照することになるので、以降、参照画像データと言う。  
20

【 0 0 2 4 】

具体的には、差分の絶対値和が最小となるブロックを探索する。参照画像データ  $S(k-1)$  内の座標 (p, q) の画素を  $P(p, q)$  と表し、r, s それぞれを 0 乃至 7 の範囲で変化させたとき、差分の絶対値和 D は、  
30

$$D = |P(p+r, q+s) - B_{i,j}(r, s)|$$

と表される。そこで、絶対値和 D が最小となる p, q を求める。ブロック  $B(i, j)$  の左上隅の画素の座標は (i × M, j × N) で表される。よって、動き補償部 104 が生成するベクトル情報 (相対位置) は、『(i × M - p, j × N - q)』となる。

【 0 0 2 5 】

なお、上記では差分の絶対値和を求めるものであったが、ブロック  $B(i, j)$  に近似する度合いを評価できればよいので、例えば次式のように 2 乗和を計算し、D の最小値となつた際の p, q を求めても良い。

$$D = \{P(p+r, q+s) - B_{i,j}(r, s)\}^2$$

【 0 0 2 6 】

また、CT や MRI での断層像の場合、隣接する断層像間のスライス間隔 (2 枚の画像の体軸方向の距離) は、DICOM にて規定されるので、符号化に先立ち、スライス間隔を示す情報を取得することは可能である。スライス間隔が小さいほど、隣り合う画像は相関が高いと考えてよいので、スライス間隔に応じて近似ブロックの探索する範囲を決めてよい。この結果、スライス間隔が小さいほど、近似ブロックの探索範囲を狭くすることで、上記最小の D を求める処理が短時間となり、符号化に係る時間短縮が期待できる。  
40

【 0 0 2 7 】

拡張ブロック生成部 105 は、動き補償部 104 からのベクトル情報に基づき、参照画像データ  $S(k-1)$  から、差分が最小となつたブロックと、当該ブロックの上、左に隣接する拡張領域の画素群で構成される拡張ブロックを、参照用拡張ブロックとして読み出  
50

す。要するに、拡張ブロック生成部 105 は、参照画像データ  $S(k-1)$  における座標  $(p-1, q-1)$  を左上隅の画素とする  $(M+1) \times (N+1)$  画素で構成される参照用拡張ブロックを読み出し、差分画像生成部 106 に供給する。実施形態では、 $M=N=8$  としているので、拡張ブロック生成部 105 は、画像データ  $S(k-1)$  から  $9 \times 9$  画素で構成される参照用拡張ブロックを読み出すことになる。なお、差分が最小となつたブロック内の左上隅の画素が、参照画像データ  $S(k-1)$  の上端や左端に位置しているとき、拡張領域の画素の一部または全部の画素が実在しない。このように実在しない画素値は、既に説明したように所定値（実施形態では“0”）であるものとみなす。

#### 【0028】

差分画像生成部 106 は、符号化対象の着目拡張ブロックと参照用拡張ブロックの各画素の画素値の差分を算出し、差分拡張ブロックデータを生成する。 10

#### 【0029】

予測符号化部 107 は、差分画像生成部 106 が生成した差分拡張画像データ内の、差分ブロックを構成する  $8 \times 8$  個の差分画素データに対して、ラスタースキャン順に予測符号化を行う。本実施形態では符号化方式として、国際標準で可逆符号化として規格化されている J P E G - L S 符号化方式（ISO / IEC 14495-1）を適用する。ただし、これに限らず、着目画素の周囲画素から予測を生成し、予測誤差を符号化するような方式であれば、J P E G の可逆符号化方式など他の符号化方式を適用しても構わない。予測符号化部 107 で符号化するのは、図 4 に示す差分拡張画像ブロックデータのうちの太枠内  $8 \times 8$  個の画素（実施形態では  $M=N=8$  としている）の画素である。太枠外の拡張領域の画素値（正確には差分画素値）については、符号化対象領域の画素を符号化する際の予測画素、あるいは、コンテキスト生成（状態判別）を行うため参照画素として利用する。この点は、符号化対象画像の周囲に、符号化側、復号側で共通の仮想的な画素を設定する J P E G - L S とは異なる。例えば、着目画素が図 4 の X に位置する場合、X の周囲 4 画素 a, b, c, d を参照して符号化する。つまり、符号化対象領域内の先頭ライン M 画素と左端一列の N 画素を符号化する際には、拡張領域の画素が利用される。J P E G - L S 符号化方式では、符号化対象領域の各画素に対して a, b, c に位置する画素から予測値  $p$  を算出し、予測値  $p$  と着目画素 X の差分値（予測誤差値  $e$ ）をエントロピー符号化する。なお、着目画素 X がブロックの右端の画素であり、かつ、ブロックの先頭ラインでない場合、周囲画素 d は未符号化ブロックに属し、利用できない。この場合は、J P E G - L S 標準と同様にして仮想的な画素を設定して用いるものとする。こうして予測符号化部 107 は、符号化対象領域  $M \times N$  画素の符号化データを生成し出力する。このとき、予測符号化部 107 は、動き補償部 104 からの動きベクトルも併せて符号列形成部 108 に供給する。 20 30

#### 【0030】

符号列形成部 108 は、復号に必要な付加情報と、予測符号化部 107 から出力される符号化データから、所定のフォーマットの符号列を形成し、信号線 110 に出力する。図 16 に符号列形成部 108 から出力される符号列の構成の一例を示す。先頭の H e a d e r 部分に付加情報が格納され、続いて各ブロックの符号化データが並ぶ。付加情報としては、画像データセットに含まれる画像データの枚数、各画像の水平方向 / 垂直方向の画素数、画素を構成する成分の数、各成分の精度、ブロックの水平方向 / 垂直方向の画素数などを格納する。なお、先頭の画像データ  $S(0)$  は先に説明したように、単独で可逆符号化したデータとなり、2 枚目以降が直前に符号化した画像を参照した可逆符号化データとなる。そして、2 枚目以降の各ブロックには、その動きベクトルデータを含むことになる。動きベクトルデータについては、符号列形成部 108 にて符号化データに含んでも構わない。例えば、H e a d e r に付加情報として格納してもよいし、ブロック符号化データの先頭にブロックヘッダとして付加してもよい。 40

#### 【0031】

なお、動き補償を行なう際に符号化対象ブロックのサイズで動きブロック探索を行うとしたが、探索のブロックサイズについてはこれに限るものではない。拡張ブロックのサイ 50

ズで動き補償を行った方が参照画素も含めた上で差分が最小となるブロックが探索できるため、符号化性能が良くなる可能性もある。そのため、拡張ブロックのサイズで動き探索を行っても構わない。また、上記説明では同一サイズのブロックに分割して動き補償を行う例について述べたが、これに限らず、異なるサイズのブロックが混在しても構わない。例えば、動きベクトル探索で得られた参照ブロックとの差分が大きい場合には着目ブロックを4分割して、より小さいブロックで動き補償を行うなどしても良い。また、本実施形態では、符号化対象ブロックに対する参照ブロック特定の方法として動き補償を行ったが、他の方法により参照ブロックを特定しても構わない。例えば、医用スライス画像の場合には撮影する体の部位によって、外形が拡張や縮小など特定の変化をする場合がある。この場合には、スライス厚さなどの情報から画像の領域毎にブロック単位に動かす方向や距離を予め設定しておき、その設定値に基づいて参照ブロックを特定してもよい。具体的には図7の(a)(b)(c)に示すように、頭部(斜線部)の外形が広がっていくよう変化する場合、太枠内が似たブロックとなることがわかる。このような例では、図8に示すように、画像をエリアに分割し、右上のエリア1に位置するブロックについては右下ブロックを参照するなど、エリアごとに参照ブロック位置を予め設定しておいてもよい。

#### 【0032】

以上の説明のように実施形態によれば、着目ブロックと参照ブロックの双方の拡張ブロックの差分を求め、その差分拡張ブロックから、符号化対象の着目ブロック内の画素を予測符号化する。この結果、着目ブロック境界において画素値の不連続性を抑制できる。つまり、差分画像の冗長性を除去しつつ、予測符号化の符号化効率を向上させることが可能となる。よって本第1の実施形態によれば、CTやMRIなどの多数のスライス画像で構成される画像データセットを、効率よく可逆符号化することが可能となる。なお、実施形態での符号化対象の画像データは、モノクロ多値であるものとして説明したが、これに限られるものではなく、他の色空間でも構わないし、1画素が12ビット以外のビット数であっても構わない。また、実施形態では、画像データセットの着目画像データS(k)を符号化する際、直前に符号化した画像データS(k-1)を参照画像として用いる例を示したが、符号化済みの画像データであれば特に直前の画像データでなくても構わない。ただし、符号化速度を高速化するためには、比較的狭い範囲の探索処理で参照ブロックを決定できることが望ましい。それ故、MRやCTなどでは、隣り合った位置のスライス画像ほど互いに高い相関を有することが期待できるので、上記実施形態のように、直前の画像データを参照画像データとすることが望ましい。色空間、ビット数、並びに、参照画像データの決め方については、以下に説明する変形例、他の実施形態でも同様である。

#### 【0033】

また、上記実施形態では、符号化対象の画像データS(k)、並びに、既符号化画像データS(k-1)におけるブロックのサイズをM×N(実施形態ではM=N=8)、拡張ブロックのサイズを(M+1)×(N+1)画素とした。これは、着目ブロックを符号化する際の予測符号化の際に参照する画素群が、図4に示すように、着目画素の上に位置する1水平ラインと、左に位置する1垂直ライン上に存在するからである。参照する画素が複数ラインに存在する場合、拡張ブロックのサイズは(M+P)×(N+Q)(P, Qは1以上の整数)と表現できる。

#### 【0034】

##### [変形例の説明]

上記第1の実施形態では、図1の構成に基づいて説明したが、上記第1の実施形態と同等の処理をパーソナルコンピュータ(以下、PC)等に代表される情報処理装置にて実行されるコンピュータプログラム(ソフトウェア)でもって実現させても構わない。図中、1401はCPUで、RAM1402やROM1403に記憶されているプログラムやデータを用いて本装置全体の制御を行うと共に、後述する画像符号化処理、復号処理を実行する。

#### 【0035】

1402はRAMである。このRAM1402は、外部記憶装置1407、記憶媒体ド

10

20

30

40

50

ライブ1408から、あるいは、I/F1409等を介して外部装置からダウンロードされたプログラムやデータを記憶するために利用される。また、このRAM1402は、CPU1401が各種の処理を実行する際のワークエリアとしても使用される。1403はROMで、ブートプログラムや本装置の設定プログラムやデータを格納する。1404、1405は夫々キーボード、マウス等のポインティングデバイスで、CPU1401に対して各種の指示を入力することができる。1406は表示装置で、CRTや液晶画面などにより構成されており、画像や文字などの情報を表示することができる。1407は外部記憶装置で、ハードディスクドライブ装置等の大容量情報記憶装置である。ここにOSや前述の画像符号化、復号処理の為のプログラム、符号化対象の画像データ、復号対象画像の符号化データなどが保存されており、CPU1401による制御によって、これらのプログラムやデータはRAM1402上の所定のエリアにロードされる。10

#### 【0036】

1408は記憶媒体ドライブで、CD-ROMやDVD-ROMなどの記憶媒体に記録されたプログラムやデータを読み出してRAM1402や外部記憶装置1407に出力するものである。なお、この記憶媒体には、オペレーティングシステム(OS)をはじめ、前述の画像符号化、復号処理の為のプログラムが格納されている。この記憶媒体に、符号化対象の画像データ、復号対象の画像の符号化データなどを記録しておいても良い。

#### 【0037】

1409はI/Fで、このI/F1409によって外部装置を本装置に接続し、本装置と外部装置との間でデータ通信を可能にするものである。例えば符号化対象の画像データや、復号対象の画像の符号化データなどを本装置のRAM1402や外部記憶装置1407、あるいは記憶媒体ドライブ1408に入力することもできる。1410は上述の各部を繋ぐバスである。20

#### 【0038】

上記構成において、本装置の電源をONにすると、CPU1401はROM1403に格納されたOSをRAM1402にロードし、起動することになる。OS起動後、本変形例における画像符号化アプリケーションプログラムをOSを介して外部記憶装置1407からRAM1402にロードし、実行する。

#### 【0039】

ここでは、説明を簡単なものとするため、インターフェース1409にCTやMRIなどの撮影装置が接続されているものとする。そして、その撮影装置で撮影されたデータ(生データ)を再構成してえられた画像データセット(例えば1000枚のスライス画像)がインターフェース1409を介して本装置に入力されるものとする。本装置は、画像データセットの構成する各画像を符号化し、外部記憶装置1407にファイルとして格納する。説明を簡単なものとするため、入力する画像セットは上記第1の実施形態と同じく、モノクロ多値の12ビット(各画素2バイトで格納)で表わされているものとする。30

#### 【0040】

図13は、画像符号化アプリケーションを実行した際の、CPU1401の処理手順を示すフローチャートである。以下、同図に従って、本変形例の符号化処理を説明する。なお、以下の説明において、画像データセット中のk番目の符号化対象の着目画像データをS(k)と表す。そして、着目画像データS(k)内の符号化対象のブロックをB(実施形態では8×8画素)、符号化対象ブロックBとその拡張領域で構成される拡張ブロックをEBと表す。そして、着目ブロックBとの差が最小となる、参照画像データS(k-1)内の参照ブロックをRB、参照ブロックRBとその拡張領域で構成される参照用拡張ブロックをREBと表す。40

#### 【0041】

まず、CPU1401は、S1にて、変数kを0で初期化する。そして、S2にて、CPU1401は、画像データS(k)(画像データセットの先頭の画像データ)を、単独で可逆符号化する。実施形態では、JPEG-LSに従って符号化するものとする。符号化して得られた符号化データは、RAM1402内に予め確保された出力バッファに格納50

する。この出力バッファが、ある程度の未書き込みデータで占められた場合、CPU1401は外部記憶媒体1407に書き込み処理を行う。以下の説明での符号化処理で生成された符号化データも同様に、この出力バッファに蓄積されるものとして説明する。

#### 【0042】

画像データセットの先頭画像データS(0)の符号化処理を終えると、S3にて、CPU1401は変数kを“1”増加させる。そして、CPU1401は、S4乃至S9のるループにて、画像データS(k)の符号化処理を行う。

#### 【0043】

まず、S4にて、CPU1401は、符号化対象の画像データS(k)から、着目ブロックを包含する拡張ブロックEBを読み出す。次いでCPU1401は、S5にて、参照画像データS(k-1)内から、拡張ブロックEB内の着目ブロックBとの差が最も小さい参照ブロックRBを探し出す。そして、CPU1401は、参照ブロックRBに対する着目ブロックBの相対座標を動きベクトルとして求める。そして、S6にて、CPU1401は、参照ブロックRBとその拡張領域で構成される参照拡張ブロックREBを、参照画像データS(k-1)から読み出す。そして、S7にて、CPU1401は、拡張ブロックEBと参照拡張ブロックREBとの差分を、差分拡張ブロックDEBとして算出する。そして、CPU1401は、S8にて、差分拡張ブロックDEBを参照して、差分ブロックDB(着目ブロックBと参照ブロックRBの差分)内の各画素の差分値を予測符号化する。このとき、符号化データの先頭には、S5で求めた動きベクトル情報が付加されるものとする。そして、CPU1401は、S9にて、着目画像データS(k)内の全ブロックの符号化を終えたか否かを判定し、否の場合には、次のブロックの符号化を行うため、S4に処理を戻す。

#### 【0044】

一方、着目画像データS(k)内の全ブロックの符号化を終えたと判断した場合、CPU1401は、S10にて、画像データセットの全画像についての符号化を終えたか否かを判断する。否の場合には、次の画像データを符号化するため、S3以降の処理を繰り返す。一方、全画像データの符号化を終えたと判断した場合には、CPU1401は、この画像符号化処理を終える。

#### 【0045】

以上説明したように、先に説明した第1の実施形態と同様の処理を、PC等の汎用の情報処理装置上で実行するアプリケーションプログラムとしても実現できることになり、第1の実施形態と同様の作用効果を得ることが可能である。

#### 【0046】

##### [第2の実施形態]

次に、上記第1の実施形態またはその変形例で示された装置で得られた符号化データを復号する画像処理装置を第2の実施形態として説明する。図15は、本第1の実施形態における復号処理を行う画像処理装置のブロック構成図である。図示のように、この画像処理装置は、符号バッファ1501、ヘッダ解析部1502、動き補償部1503、周辺画素設定部1504、差分画像復号部1505、原画像ブロック復号部1506、画像形成部1507とを備える。同図において1508、1509は信号線である。

#### 【0047】

以下、図15を参照して、本実施形態に係る画像処理装置が行う復号処理について説明する。なお、画像データセットの先頭画像S(0)は、JPEG-LSに従って既に復号済みであるものとする。以下の説明は、先頭の画像データS(0)以外の、k番目(k1)の画像データS(k)の復号処理である。

#### 【0048】

信号線1508を介して、復号対象となる符号化データとこの復号対象符号化データに対応する動きベクトルデータが本画像処理装置に入力される。符号化データの供給源は特に問わないが、ここではメモリカードなどの記憶媒体として説明する。

#### 【0049】

10

20

30

40

50

ヘッダ解析部 1502 は符号バッファ 1501 に格納される符号化データのヘッダ部分を解析する。そして、ヘッダ解析部 1502 は、画像の水平方向 / 垂直方向の画素数、ブロックの水平方向 / 垂直方向の画素数、画素を構成する成分の数、各成分の精度（ビット数と有効ビット数）などの付加情報を得る。これらの付加情報は復号の過程で利用されるものとする。

#### 【0050】

動き補償部 1503 は既に復号済みの参照画像データ  $S(k-1)$  と、動きベクトルデータを受け取る。そして、動き補償部 1503 は、参照画像データ  $S(k-1)$  から、動きベクトルで示される参照ブロック  $RB$  を包含する参照拡張ブロック  $REB$  を読み出し、差分画像復号部 1505 に供給する。

10

#### 【0051】

周辺画素設定部 1504 は、参照拡張ブロック  $REB$  内の拡張領域の画素と、復号対象の画像データ  $S(k)$  の着目ブロック  $B$  の拡張領域の画素を読み出し、それらの差分を算出する。そして、算出した拡張領域の各画素の差分値を、着目ブロック  $B$  の差分画像を復号する際の周辺画素として、差分画像復号部 1505 に設定する。復号対象ブロック  $B$  が画像端に位置するブロックの場合には、復号済みの隣接画素が存在しない位置の画素について符号化時に設定したのと同じ初期値を設定し、差分を生成する。

#### 【0052】

具体的な例を図 6 に示し説明する。図 6 (a) は参照拡張ブロック  $REB$  の例を示す。図 6 (a) の太枠内が動き補償部 1503 にて特定された参照ブロック  $RB$  ( $8 \times 8$  画素) であり、参照ブロック  $RB$  に隣接する上 1 ラインと左 1 列の画素が拡張領域の画素である。また、図 6 (b) に、着目画像データ  $S(k)$  の着目ブロック  $B$  の例を示す。図示の太枠が復号対象の着目ブロック  $B$  であり、まだ復号されていないため画素値はこの段階では不明である。太枠外の格子パターンで示されるのが、拡張領域の画素であり、これらは既に復号済みである。よって、差分画像復号部 1505 は、図 6 (a) の拡張領域の画素と、図 6 (b) の拡張領域の画素との差分を演算し、その結果（図 6 (c) の斜線部分の画素）を、復号対象の着目ブロック  $B$  の周辺画素値として設定する。

20

#### 【0053】

差分画像復号部 1505 は、周辺画素設定部 1504 により設定された周辺画素と、復号対象ブロックの符号化データとを受け取り、復号対象の着目ブロックの復号処理を行う。着目ブロックの左上隅の画素を予測復号する場合、その予測値は図 4 に示すように、画素  $a$ ,  $b$ ,  $c$  を必要とする。本実施形態の場合、画素  $a$ ,  $b$ ,  $c$  は、周辺画素設定部 1504 により設定された周辺画素を参照すればよいので、予測値は正しく算出できる。着目ブロックの左上隅の画素を除く先頭ラインの画素も同様に正しく復号できる。

30

#### 【0054】

なお、差分画像復号部 1505 から生成されるデータは、着目ブロック  $B$  内の画素と、参照ブロック  $RB$  内の画素との差分画素データである。それ故、原画像ブロックは、着目ブロック  $B$  の差分画素データそれぞれに、参照ブロック  $RB$  の各画素を加算することで、着目ブロック  $B$  の画像データを生成し、生成した着目ブロック  $B$  の画像データを画像系シエーディング部 1507 に供給する。

40

#### 【0055】

画像形成部 1507 は、復号された全てのブロックデータを画像データとして構成し、信号線 1509 により出力する。出力対象は、ハードディスク等の外部記憶装置とするが表示装置であっても良く、その種類は問わない。復号済みの画像データ  $S(k)$  は、その次の画像データを復号する際の、参照画像データとして利用されることになる。以上の動作により、符号化データから完全に元の画像データを復元することができる。

#### 【0056】

以上説明したように、本第 2 の実施形態によれば、第 1 の実施形態、並びに、その変形例で示した装置で生成された符号化データから、画像データセットの各画像を復号することができる。なお、本第 2 の実施形態での、復号処理は、先に説明した第 1 の実施形態の

50

変形例と同様に、コンピュータプログラムでもって実現できるのは明らかである。この場合、その復号プログラムは、図15の各機能部に相当する関数やサブルーチンを有することで実現できる。

【0057】

【第3の実施形態】

第1の実施形態では、符号化対象ブロックと、そのブロックに隣接し、既符号化の画素を含めた拡張ブロックと参照拡張ブロックとの差分を求めて、予測符号化する方法について説明を行った。

【0058】

この場合、ブロック端の差分画素を符号化する場合に、拡張領域の差分画素を参照画素として利用できるため、符号化効率を向上できる。しかしながら、符号化対象となる差分画像ブロックを符号化する際のコンテキスト情報として、符号化済みブロックで生成された情報を利用して予測符号化している。この場合には、必ずしも符号化対象の差分画像ブロックと、これまでに符号化したブロックの特性が近いとは限らず、適切な確率モデルの推定ができていない可能性がある。そこで、本第3の実施形態では、符号化対象ブロックに隣接する、より広い既符号化ブロック内の画素領域を含む拡張ブロックを生成し、拡張領域の画素について、符号化対象ブロック用のコンテキスト生成を行う方法について以下に説明を行う。

10

【0059】

なお、本第3の実施形態に係る画像処理装置の符号化対象とする画像データは、第1の実施形態と同様に、グレイスケール画像データであり、1コンポーネント12ビットで0～4095の範囲の輝度値を表現した画素データにより構成されるものとする。以下の説明に成分値という表現が出てくるが、本実施形態においては1成分のみが入力されるため、成分値=画素値として説明を行う。また、画像は水平方向W画素、垂直方向H画素により構成される点も第1の実施形態と同じとする。

20

【0060】

本第3の実施形態に係る画像処理装置のブロック図は、第1の実施形態で説明した図1のブロック構成図と同じであるため、新たに図示しない。本実施形態において、第1の実施形態と異なるのは、拡張ブロック生成部105と予測符号化部107の動作のみであるので、動作の異なる拡張ブロック生成部105と予測符号化部107の処理の流れについて以下に説明を行う。

30

【0061】

まず、本実施形態に係る拡張ブロック生成部105は拡張領域の画素として符号化対象ブロックBと隣接する上1ラインと左1列のみでなく、任意の範囲の領域を設定する。例えば、図5に示す斜線部分のように、着目ブロックの上に位置する数ラインおよび左に位置する数列を拡張領域とする。本第3の実施形態では、符号化対象領域の左上、上、左に位置する領域の画素（図5の斜線部分）を拡張領域の画素とした。しかし、図5の二重線で示す枠内エリア1の上方数ラインのみや、左側数列（図5のエリア2）のみ、左上（図5のエリア3）のみを拡張画素としても構わない。また、符号化済み画素や符号化対象領域の画素配置から、ブロック毎に拡張画素のエリアを切り替えて構わない。ここでは説明を単純なものとするため、図5に示すように符号化対象ブロックBが8×8画素のときに、それを包含する12×12画素を拡張ブロックとする。

40

【0062】

続いて、本第3の実施形態に係る予測符号化部107の符号化処理について以下に説明する。本第3の実施形態においても、予測符号化として第1の実施形態と同様に、JPEG-LS符号化方式を適用するものとする。ただし、これに限らず、コンテキストモデルを利用して符号化するような方式であれば他の符号化方式を適用しても構わない。第1の実施形態では、符号化対象領域についてのみ予測符号化処理を行ったが、本実施形態においては拡張領域の画素についてもJPEG-LSにて予測符号化を行う。

【0063】

50

具体的な方法について図5を用いて説明する。まず、斜線で示す拡張領域の画素について、ラスタースキャン順に予測符号化処理を行う。JPEG-LSでは着目画素Xの周囲4画素a, b, c, dから、画素ペアa, cの差分、画素ペアc, bの差分、画素ペアb, dの差分(画素勾配とも言う)から365種類のコンテキストを生成する。コンテキストによって、予測誤差値eの補正值やGolomb符号により予測誤差値eを符号化する際に用いるパラメータ変数kの値を決定する。Golomb符号は非負の整数値を符号化対象とし、パラメータ変数kによって異なる複数の確率モデルによる符号化が可能であるという特徴を持つ。JPEG-LS符号化方式でのコンテキスト生成やGolomb符号化の詳細な方法については、勧告書に記載があるため、ここでは説明を省略する。

## 【0064】

10

拡張領域の画素について予測符号化処理を終了した時点でのコンテキスト情報を、符号化対象領域の画素を符号化する際に用いる。ここで、拡張領域の画素については、更新されたコンテキスト情報のみ用いられ、符号化データについては不要のため出力しない。なお、本実施形態においては、拡張領域について符号化処理を行ったうえでコンテキスト情報のみ利用するものとしたが、拡張領域についてはGolomb符号化の処理を省略しても構わない。

## 【0065】

20

以上の説明のように、隣接領域の画素についてコンテキストを生成し、符号化対象画素の符号化に利用することで、符号化対象ブロックの特性にあった確率モデルの推定が可能となり、更なる符号化効率の向上が実現できる。

## 【0066】

## [第4の実施形態]

第3の実施形態においては、拡張領域について生成したコンテキスト情報(以下、単にコンテキストと)を用いて予測符号化する例について説明を行った。本第4の実施形態では拡張領域についてコンテキスト生成する際に得られる情報を基に、符号化パラメータを選択する方法について説明を行う。

## 【0067】

30

なお、本第4の実施形態に係る画像処理装置の符号化対象とする画像データは、第1の実施形態と同様に、グレイスケール画像データであり、1コンポーネント12ビットで0~4095の範囲の輝度値を表現した画素データにより構成されるものとする。また、画像は水平方向W画素、垂直方向H画素により構成されるものとする。

## 【0068】

本第4の実施形態に係る画像処理装置のブロック図は、第3の実施形態で説明した図1のブロック構成図と同じであるため、新たに図示しない。本第4の実施形態において、第1の実施形態と異なるのは、予測符号化部107の動作のみであるので、動作の異なる予測符号化部107の処理の流れについて以下に説明を行う。

## 【0069】

40

第3の実施形態では、主にkパラメータの更新方法について説明を行った。本実施形態の符号化処理の説明の前に、JPEG-LSでのコンテキスト生成の詳細な方法について説明しておく。JPEG-LSでは着目画素の周囲4画素a, b, c, dの画素勾配から、コンテキストを365状態に分け、各コンテキストに対するコンテキストパラメータの更新を行う。具体的には、まず差分値d1, d2, d3を生成する。各差分値は、d1 = d - b, d2 = b - c, d3 = c - aで求められる。次に、差分値d1, d2, d3を各々-4~4までの9値に量子化する。量子化の閾値をT1, T2, T3として、図9に示すように、入力dに対して出力Quantize(d)が決まる。差分値d1, d2, d3を量子化して得た値をD1, D2, D3と表現するものとする。

## 【0070】

50

通常、ブロック単位に符号化する際には、画像に対してコンテキスト365状態を1セットとして保持して、ブロック毎に生成するか、符号化済みブロックで更新したコンテキスト情報を着目ブロックで利用するかの方法で用いることが考えられる。1枚の画像データ

タに対して、1セットのコンテキスト情報を利用する場合には、画像内で特性の異なる領域が混在している場合に、領域毎に適したコンテキストが適用されず、符号化効率が低下する可能性がある。これを解決するために、ブロックなどの決められた領域ごとにコンテキストを生成し、符号化済みの画像のブロック毎のコンテキスト情報を符号化対象の画像データのブロックに適用する方法も考えられる。この場合には、複数のコンテキストセットを保持する必要があり、符号化や復号処理の際にメモリ使用量が膨大になってしまう。例えば、 $512 \times 512$  の画像データを  $8 \times 8$  サイズのブロックに分割して、各ブロックに対してコンテキストを生成した場合には、 $64 \times 64 = 4096$  セットのコンテキスト情報を保持することになる。

## 【0071】

10

本実施形態において、予測符号化部 107 は拡張領域の画素配置から、符号化対象領域を符号化する際に適用するコンテキスト情報を選択する。例えば、拡張領域における画素値のヒストグラムを基に平坦な領域であるか、エッジの多い領域であるかなどを判断する。領域のタイプ分類の方法については、ヒストグラムの取得に限らず、エッジ検出やコンテキスト生成を行うなど、どんな方法を用いても構わない。

## 【0072】

本実施形態においては、拡張領域について予測誤差の頻度分布を生成し、このヒストグラムから画像のタイプを選択する。図 10 に示すタイプ A、B、C の 3 種類を設ける場合の例について、以下に説明する。図 11 には、ヒストグラムの例を示す。

## 【0073】

20

図 11 (a) に示すようなヒストグラムのブロックは平坦領域であり、タイプ A を選択する。コンテキストセットは CS1、コンテキスト生成のための量子化閾値は T1\_a、T2\_a、T3\_a とする。平坦領域については予測誤差値が 0 に集中するため、JPEGSLS 符号化方式のデフォルトの量子化閾値を利用すると、発生するコンテキストが偏る可能性ある。そのため、出現する予測誤差値の範囲内に閾値を設定して、365 状態を有効利用できるようにする。

## 【0074】

コンテキストセットについては、予めタイプに適したコンテキスト情報を用意しておいて固定値のコンテキストセットを適用してもよいし、これまでに出現したタイプ毎にコンテキストを更新してもよい。また、タイプ毎に適していると思われる初期値を設定しておいて、その値から符号化時にコンテキストの更新を行っても構わない。

30

## 【0075】

続いて、図 11 (b) に示すようなヒストグラムのブロックは自然画領域であり、タイプ B を選択する。この場合もタイプ A と同様に出現した予測誤差値のレンジから、閾値 T1\_b、T2\_b、T3\_b を設定すればよい。図 11 (c) に示すようなヒストグラムのブロックはエッジ領域であり、タイプ C を選択する。コンテキストセットは CS3、量子化閾値は T1\_c、T2\_c、T3\_c を選択する。

## 【0076】

以上の説明のように、拡張領域の予測誤差の頻度分布から、適したコンテキストセットやコンテキスト生成時の量子化閾値を選択することで、メモリ使用量を抑制しつつ、符号化効率を向上できる。

40

## 【0077】

## [第 5 の実施形態]

本実施形態は、符号化対象ブロック B に隣接する既符号化のブロックの動きベクトルによって、符号化済みブロックで生成したコンテキストを利用する、符号化対象ブロックの拡張領域の画素で生成したコンテキスト情報を利用するかを切り替える。

## 【0078】

まず、符号化対象ブロック B の動きベクトルと、それに符号化済みの隣接ブロックの動きベクトルの比較を行う。動きベクトルの距離や方向を比較した結果、ブロック間の相関が高いと判断された場合には、符号化済み隣接ブロックにて生成したコンテキストを利用す

50

る。一方、ブロック間の相関が低い状態であると判断された場合には、符号化対象ブロックの拡張領域の画素でコンテキストを生成し、そのコンテキストを利用するものとする。

#### 【0079】

図12を参照して、より分かりやすく説明する。着目ブロックBxと、符号化済みの隣接ブロックBa、Bb、Bc各ブロックの動きベクトルが図12に示すような方向であった場合、各ブロックに対する参照ブロックは空間的に離れた位置のブロックである。そのため差分画像ブロックを比較した場合に、画素配置の特性が異なる可能性がある。ブロック内の画素配置の特性が異なれば、隣接ブロックで生成したコンテキスト情報をを利用して予測符号化を行うことで、符号化効率が低下することが考えられる。このような場合には、隣接ブロックを符号化する際に生成したコンテキストではなく、符号化対象ブロックの拡張領域について生成したコンテキストを利用して符号化する。

10

#### 【0080】

具体的には、着目ブロックの動きベクトルをVx、ブロックBa、Bb、Bcの動きベクトルをVa, Vb, Vc、閾値をThと定義したとき、次式が満足するとき、着目ブロックBxについては、拡張領域について生成したコンテキストを利用して符号化する。

|Vx - Va| > Th かつ

|Vx - Vb| > Th かつ

|Vx - Vc| > Th

ここで|v|は、ベクトルvの大きさ(スカラー)を表すものとする。

#### 【0081】

20

一方、上記の比較結果、隣接ブロックと符号化対象ブロックの動きベクトルの方向や距離が近ければ、隣接ブロックを符号化する際に生成したコンテキストをそのまま利用する。ブロック単位に利用するコンテキストセットを切り替える場合には、ブロックごと独立でコンテキスト生成してもよいし、隣接ブロックのコンテキストを利用するブロックでは、利用したコンテキストセットを更新しても構わない。また、符号化済みの隣接ブロックBa、Bb、Bcのうち、1つでも近い動きベクトルを持つブロックがあれば、近い動きベクトルを持つブロックで生成されたコンテキストを利用して構わない。

#### 【0082】

以上の説明のように、隣接ブロックの動きベクトルと、符号化対象ブロックに対する動きベクトルの比較を行い、ベクトル方向が異なる場合には、拡張領域でコンテキスト生成を行い、符号化対象ブロックの符号化に利用する。この判定を行うことで、適切なコンテキスト情報を符号化対象ブロックに対して適用できる。

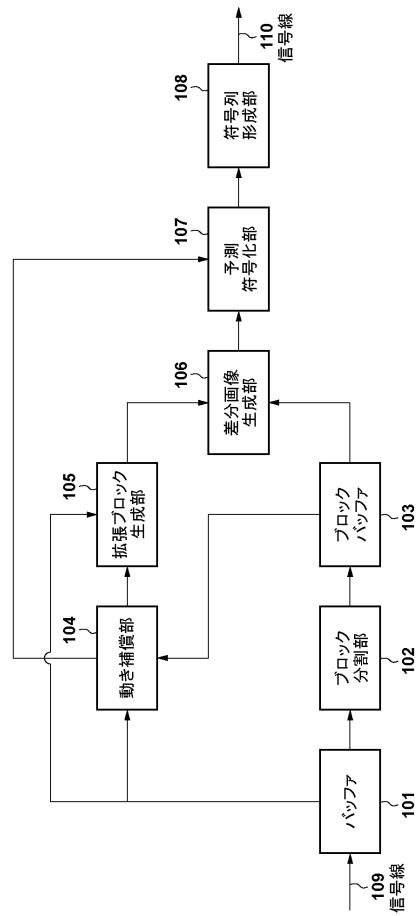
30

#### 【符号の説明】

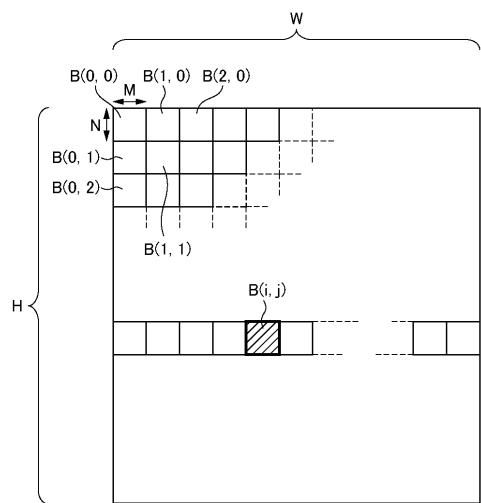
#### 【0083】

101...バッファ、102...ブロック分割部、103...ブロックバッファ、104...動き補償部、105...拡張ブロック生成部、106...差分画像生成部、107...予測符号化部、108...符号列形成部

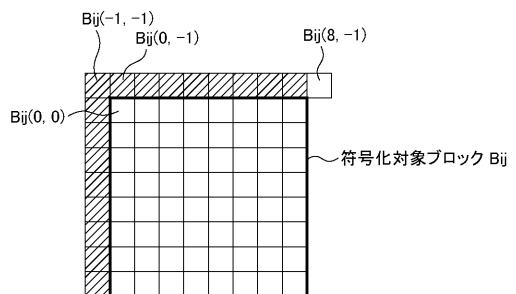
【図1】



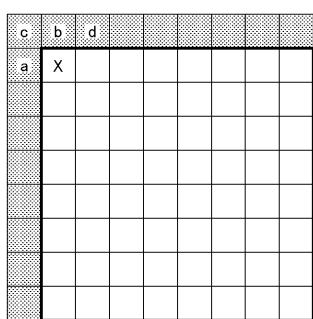
【図2】



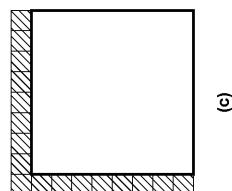
【図3】



【図4】

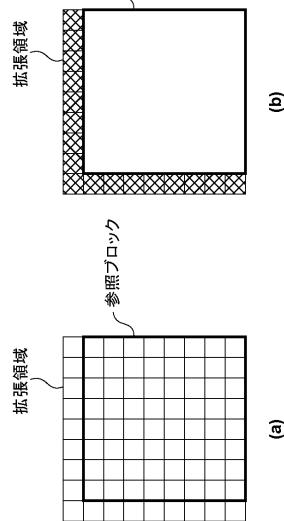
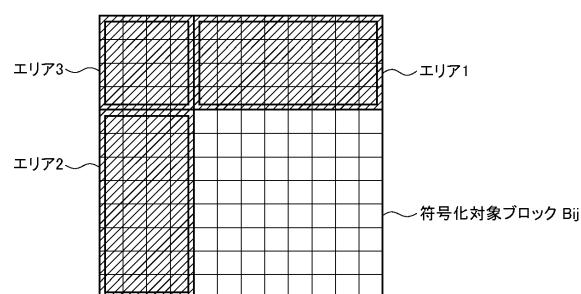


【図6】



(c)

【図5】

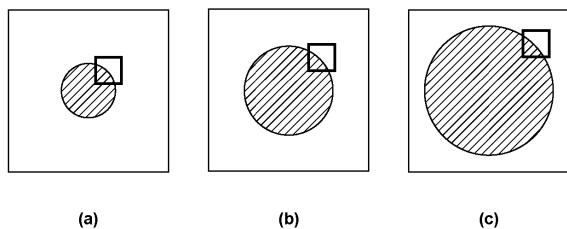


(a)

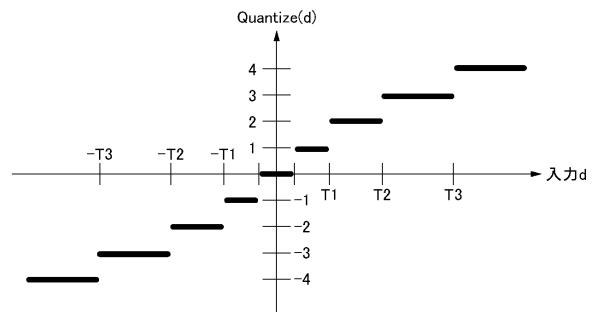
(b)

(c)

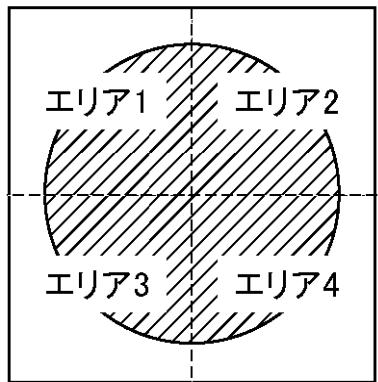
【図7】



【図9】



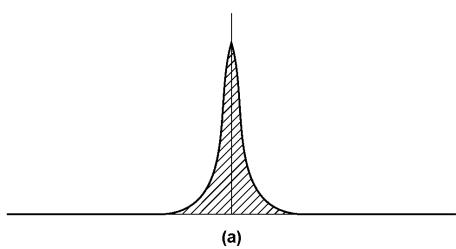
【図8】



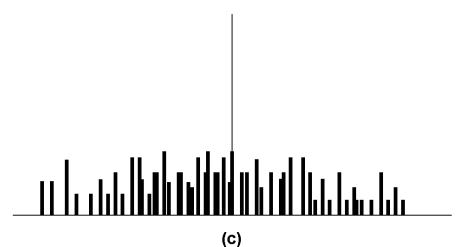
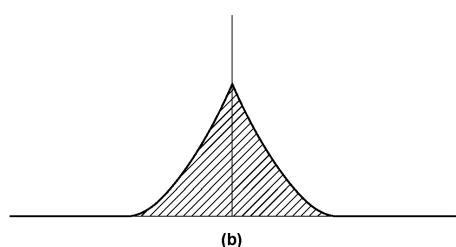
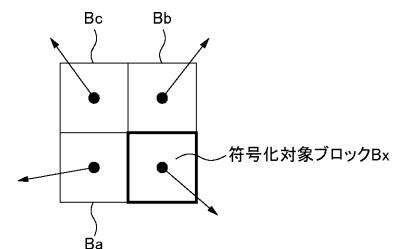
【図10】

タイプ	画像	コンテキストセット	量子化閾値
A	平坦領域	CS1	T1_a, T2_a, T3_a
B	自然画領域	CS2	T1_b, T2_b, T3_b
C	エッジ領域	CS3	T1_c, T2_c, T3_c

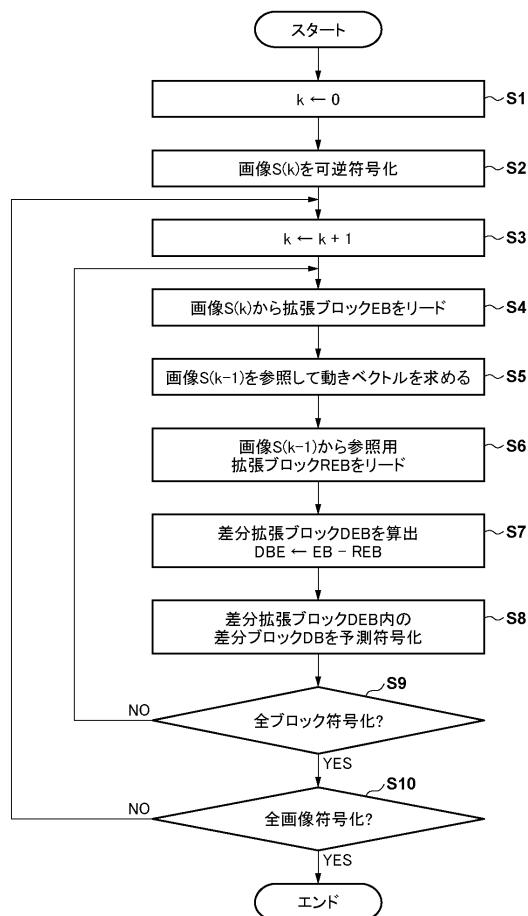
【図11】



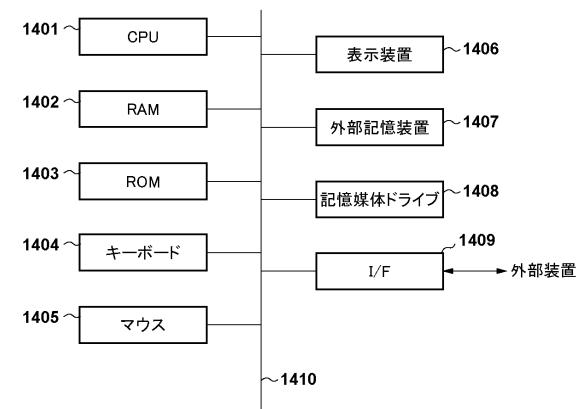
【図12】



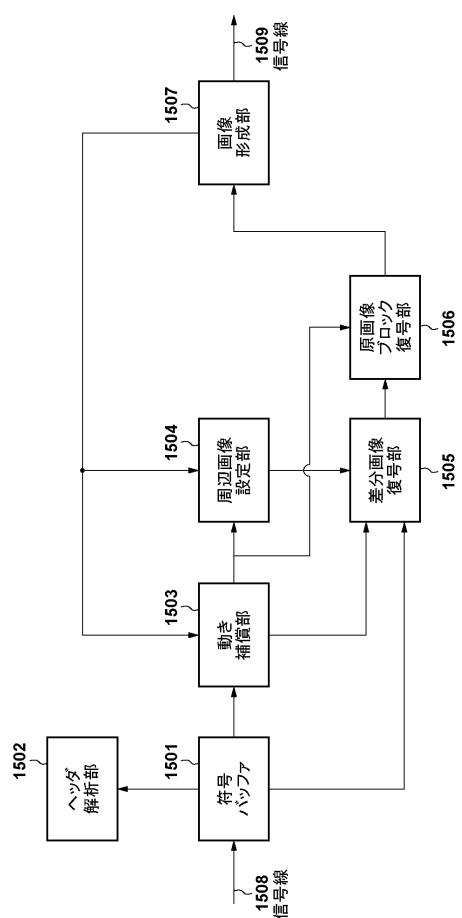
【図 1 3】



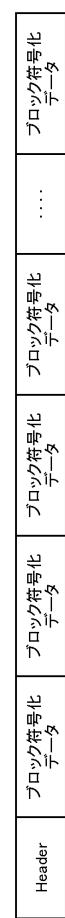
【 図 1 4 】



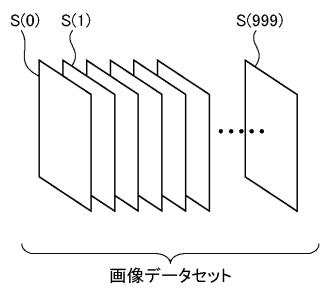
【図15】



【 図 1 6 】



【図 1 7】



---

フロントページの続き

(72)発明者 白石 友希

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 鉢呂 健

(56)参考文献 米国特許出願公開第2010/0329342(US, A1)

特表2013-511874(JP, A)

特開2010-258739(JP, A)

米国特許出願公開第2010/0310184(US, A1)

(58)調査した分野(Int.Cl., DB名)

H04N 19/00 - 19/98

IEEE Xplore