(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2008-205790
(P2008-205790A)
(43)公開日　平成20年9月4日(2008.9.4)

(51)Int.Cl.			F I			テーマコード（参考）
H04N	7/32	(2006.01)	H04N	7/137	Z	5C059
H04N	7/30	(2006.01)	H04N	7/133	Z	

審査請求 未請求　請求項の数 8　ＯＬ　（全 22 頁）

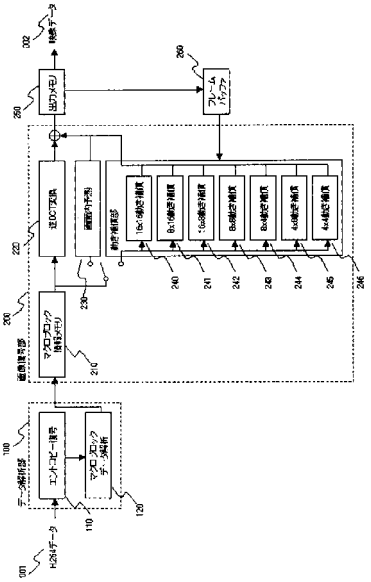
(21)出願番号　特願2007-39256(P2007-39256)
(22)出願日　平成19年2月20日(2007.2.20)

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真１００６番地
(74)代理人　100097445
弁理士　岩橋　文雄
(74)代理人　100109667
弁理士　内藤　浩樹
(74)代理人　100109151
弁理士　永野　大介
(72)発明者　竹田　真也
大阪府門真市大字門真１００６番地　松下電器産業株式会社内

Ｆターム(参考)　5C059 MA00 MA04 MA05 MA12 MA23 MC11 NN01 PP16 SS20 TA12 TA23 TA62 TB07 TC04 TC12 TC24 TC41 TC42 UA05

(54)【発明の名称】Ｈ．２６４復号装置、プログラム、および記録媒体

(57)【要約】
【課題】Ｈ．２６４の動き補償処理において、４ｘ４画素単位で生成されるダイレクトモーションベクトルの処理は、ＰＣに搭載されているＣＰＵの能力を十分に発揮できない。ＣＰＵの能力を十分に発揮させるためには、より大きな画素サイズで動き補償処理を行い、同じ演算結果を得る必要がある。
【解決手段】ダイレクトモーションベクトルの算出過程において、算出過程で参照するＣｏ－ｌｏｃａｔｅｄマクロブロックのマクロブロックタイプを用いて、隣接ベクトルの同一性を検知する手段を備えたマクロブロック解析装置と、７種類のサイズ毎にＳＳＥ２、MMXで速度最適化された動き補償演算装置を有し、ダイレクト予測処理を高速に行うことを可能にする、Ｈ．２６４復号装置、プログラムおよび媒体。
【選択図】図１



【特許請求の範囲】
【請求項１】
符号データを入力する入力ステップと、前記入力ステップで入力された符号データのマクロブロック情報解析において、画面内予測、または動き補償を行う画素単位でマクロブロックの種別を判定する判定ステップと、前記判定ステップで判定されたマクロブロックの種別に応じて、符号データから画面内予測モードとモーションベクトルと逆ＤＣＴ変換係数を取得するデータ取得ステップと、特定の画素単位で動き補償演算する複数の動き補償演算ステップと、前記判定ステップで判定されたマクロブロックの種別がインターマクロブロックの場合に、前記動き補償演算ステップで動き補償演算する画素単位を大きくするように切り替えて動き補償を行い、画像を復号する画像復号ステップとを有することを特徴とする復号方法。
【請求項２】
前記判定ステップは、空間ダイレクトモーションベクトルを有するマクロブロックのマクロブロック種別を、空間ダイレクトモーションベクトル算出過程で得られる基準モーションベクトルの大きさ、参照ピクチャ番号、あるいは、空間ダイレクトモーションベクトル算出過程で参照する他のピクチャ内のマクロブロックの種別を利用して、より大きなサイズで動き補償が可能になるように決定する第１の決定ステップを有することを特徴とする、請求項１に記載の復号方法。
【請求項３】
前記判定ステップは、時間ダイレクトモーションベクトルを有するマクロブロックに対して、マクロブロック種別を、時間ダイレクトモーションベクトル算出過程で参照する他のピクチャ内のマクロブロックの種別を利用して、より大きなサイズで動き補償が可能になるように決定する第２の決定ステップとを有することを特徴とする、請求項１に記載の復号方法。
【請求項４】
前記動き補償演算ステップは、演算器固有の拡張命令により、複数の成分を同時に演算することを特徴とする、請求項２または請求項３に記載の復号方法。
【請求項５】
Ｈ．２６４を復号することを特徴とする、請求項４に記載の復号方法。
【請求項６】
請求項１から請求項５に記載した復号方法で符号データを復号する復号装置。
【請求項７】
請求項１から請求項５に記載した復号方法を実行させるプログラム。
【請求項８】
請求項７に記載したプログラムを記録した媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ＰＣ上で動作するソフトウェアで、圧縮された画像を復号する技術に関する。
【背景技術】
【０００２】
近年、パーソナルコンピュータ（以下ＰＣ）を用いて、映像の編集が盛んに行われており、ビデオカメラで撮影した映像をＰＣ上で確認し、編集、保存するという機能を持つアプリケーションが数多く存在する。
【０００３】
一方、映像分野ではハイビジョン化が進み、ＨＤ解像度のＭＰＥＧ－２コーデックを用いた高画質な映像がビデオカメラ等で作成できるようになった。高解像度で高画質な映像を実現するには、高ビットレートでのＭＰＥＧ－２エンコードが必要となる。そこで、新たな動画コーデックであるＨ．２６４（ＭＰＥＧ－４　ＡＶＣ）に対する期待が高まって

いる。Ｈ．２６４は従来のＭＰＥＧ－２やＭＰＥＧ－４と比較し、２倍の圧縮効率を実現することが出来ると言われている。

【０００４】

しかしながら、Ｈ．２６４は圧縮率を向上させるための様々な手法が盛り込まれており、エンコードやデコードに膨大な処理を必要とする。その中でもデコード処理は、動画の再生時にはリアルタイムで行う必要がある。高解像度で高画質のＨ．２６４に対し、従来から盛んに行われているＰＣ上での動画編集を実現する際には、内容確認のためのリアルタイム再生機能を実現しなければならない。そのためには、高速に処理を行うＨ．２６４デコーダが必要となる（例えば、非特許文献１参照。）。

【０００５】

ＭＰＥＧ２規格やＨ．２６４規格の動画像データの圧縮の手法においては、動き検出と呼ばれる技術が用いられている。そして動き検出の逆処理に相当する動き補償処理と呼称される処理を行って画像を復号する。動き検出とは簡単に言うと、連続するフレーム間で写っている要素がどのように動いたかを検出して効率よく圧縮する方法である。ＭＰＥＧ２規格の動き検出においてはマクロブロック（例えば１６画素×１６画素のブロック）ごとに、そのマクロブロックが前後のフレームにおいて、どの方向へどの程度動いているかという動きベクトルを求める。そして動きベクトルと、それに対応するマクロブロックの各画素との差分データを求める。そして動き補償処理において画像を復元する際には、この動きベクトルと画素の差分データを合成して元の画像を再現する。差分データは本来の画素データと比較するとデータサイズがかなり小さくなるので、高圧縮が可能となっている。圧縮前に比して、データサイズを小さくできることから、データ転送の速度が、圧縮しない場合に比べて、かなり速くなるという効果を得られる。

【０００６】

また、Ｈ．２６４規格においてはマクロブロックには１６×１６画素、８×１６画素、１６×８画素、８×８画素の画素サイズが存在する。更には、８×８画素サイズのマクロブロックの場合には、８×４画素、４×８画素、４×４画素のサブマクロブロックに分かれる場合がある。なお、サブマクロブロックには８×８画素のものもある。この７種類の動き補償処理に対し、全てのサイズを４×４画素サイズの集合体として、１つの動き補償処理装置で処理することで、画像復号装置の作成工数や、記憶容量の圧迫を軽減させる手法などが考案されている（例えば、特許文献１参照。）。

【非特許文献１】ＩＴＵ－Ｔ（ＩｎｔｅｒｎａｔｉｏｎａｌＴｅｌｅｃｏｍｍｕｎｉｃａｔｉｏｎｓＵｎｉｏｎ－Ｔｅｌｅｃｏｍｍｕｎｉｃａｔｉｏｎｓｔａｎｄａｒｄｉｚａｔｉｏｎｓｅｃｔｏｒ）Ｈ．２６４規格書

【特許文献１】特開２００６－３１１５２６号公報

【発明の開示】

【発明が解決しようとする課題】

【０００７】

しかし、ＰＣに搭載されているＣＰＵ上で演算するソフトウェアにおいては、処理速度の観点からは、ＣＰＵの能力を十分に発揮させることができないという問題がある。そこで、本発明は、上記問題に対し、ＰＣ上でより高速に動作する画像復号装置を提供することを目的とする。

【０００８】

ＰＣに搭載されるＣＰＵには、画像処理等を高速に行うための、ＭＭＸやＳＳＥ２などのＣＰＵ拡張命令セットが用意されている。ＰＣ上で動作する通常のプログラムは、３２ｂｉｔ汎用レジスタを複数個使用することで動作する。一方、ＣＰＵ拡張命令セットを用いたプログラムは、３２ｂｉｔ汎用レジスタとは別に、複数個の６４ｂｉｔおよび１２８ｂｉｔ拡張レジスタを使用することができる。

【０００９】

また、この拡張レジスタを用いた命令セットには、レジスタ内を連続した８ｂｉｔ／１６ｂｉｔのデータの集まりとして、それぞれの値に対して、同時に同じ演算を適用するこ

とができるものがある。例えば、２つの１２８ｂｉｔレジスタを１６ｂｉｔの８個の変数の集まりとして、１６ｂｉｔ単位で加算する命令セットなどがある。このＣＰＵ拡張命令セットを有効に利用することで、ＰＣ上での動き補償を高速に行うことが可能となる。

【００１０】

さらに動き補償処理において、ＣＰＵ拡張命令セットを有効に活用にするには、より大きな画素サイズで動き補償処理を行い、同じ演算結果を得る必要がある。

【課題を解決するための手段】

【００１１】

上記課題を解決するため、本発明では、符号データを入力する入力ステップと、入力ステップで入力された符号データのマクロブロック情報解析において、画面内予測、または動き補償を行う画素単位でマクロブロックの種別を判定する判定ステップと、前記判定ステップで判定されたマクロブロックの種別に応じて、符号データから画面内予測モードとモーションベクトルと逆ＤＣＴ変換係数を取得するデータ取得ステップと、ＭＭＸ、ＳＳＥ２の拡張命令セットを用いて速度最適化した１６×１６画素、８×１６画素、１６×８画素、８×８画素、８×４画素、４×８画素、４×４画素単位で動き補償演算する７種類の動き補償演算ステップと、判定ステップで判定されたマクロブロックの種別がインターマクロブロックの場合に、動き補償演算ステップで動き補償演算する画素単位を大きくするように切り替えて動き補償を行い、画像を復号する画像復号ステップとを有することを最大の特徴とする。

【発明の効果】

【００１２】

これにより、４×４画素単位で生成されるダイレクトモーションベクトルを複数同時に動き補償処理をすることができ、より高速に映像の復号処理を実現することが可能となる。

【発明を実施するための最良の形態】

【００１３】

（実施の形態１）

図１は、本発明におけるＨ．２６４復号処理装置全体の構成を示している。データ解析部１００では、Ｈ．２６４符号データ００１を入力とし、エントロピー復号装置１１０で可変長符号を復号し、マクロブロック情報解析装置１２０により解析した情報をマクロブロック情報メモリ２１０に格納する。

【００１４】

マクロブロックデータ解析装置は、図２に示すように、ｍｂ＿ｔｙｐｅ取得部１２１で得られたｍｂ＿ｔｙｐｅにより、以降の取得するデータを決定する。Ｉ＿ＰＣＭマクロブロックの場合は、Ｉ＿ＰＣＭ情報取得部１２２において、非圧縮データを取得し、イントラマクロブロックの場合は、画面内予測モード取得部１２３、および逆ＤＣＴ変換係数取得部１２６において、画面内予測モードと逆ＤＣＴ変換係数を取得する。また、インターマクロブロックの場合は、モーションベクトル取得部１２４または、ダイレクトモーションベクトル解析部により、モーションベクトルを取得し、逆ＤＣＴ変換係数取得部１２６において、逆ＤＣＴ変換係数を取得する。これらの解析結果をマクロブロック情報メモリ２０１に格納する。

【００１５】

マクロブロック情報メモリ２０１に格納される情報には、以下の要素が含まれている。

マクロブロックタイプ

マクロブロックタイプには、画面内予測および動き補償を行う上で予測単位と用いる画素のサイズによって、８種類に分類される。

－Ｉ＿ＰＣＭ

－Ｉｎｔｒａ＿１６ｘ１６

－Ｉｎｔｒａ＿８ｘ８

－Ｉｎｔｒａ＿４ｘ４

－Ｉｎｔｅｒ＿１６ｘ１６
－Ｉｎｔｅｒ＿１６ｘ８
－Ｉｎｔｅｒ＿８ｘ１６
－Ｉｎｔｅｒ＿８ｘ８
Ｉｎｔｅｒ＿８ｘ８の場合には、さらに以下の４種類のサブマクロブロックタイプを４つ保持する。

【００１６】
－Ｉｎｔｅｒ＿８ｘ８
－Ｉｎｔｅｒ＿８ｘ４
－Ｉｎｔｅｒ＿４ｘ８
－Ｉｎｔｅｒ＿４ｘ４
・モーションベクトル
モーションベクトルは、マクロブロックタイプに応じて、以下の情報を複数個保持する。

【００１７】
－前方参照ピクチャインデックス
－前方参照モーションベクトルの水平成分
－前方参照モーションベクトルの垂直成分
－後方参照ピクチャインデックス
－後方参照モーションベクトルの水平成分
－後方参照モーションベクトルの垂直成分
マクロブロックタイプが、Ｉ＿ＰＣＭ、Ｉｎｔｒａ＿１６ｘ１６、Ｉｎｔｒａ＿８ｘ８、Ｉｎｔｒａ＿４ｘ４の場合、および前方または後方参照を行わないＩｎｔｅｒ＿ＹＹｘＺＺは、ピクチャインデックスに－１を、モーションベクトルの水平成分に０を、モーションベクトルの垂直成分に０を持つ。

【００１８】
・画面内予測モード
マクロブロックタイプが、Ｉｎｔｒａ＿１６ｘ１６の場合は、Ｈ．２６４規格で定義されるＩｎｔｒａ１６ｘ１６ＰｒｅｄＭｏｄｅおよびｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅを保持し、Ｉｎｔｒａ＿８ｘ８の場合は、Ｈ．２６４規格で定義されるＩｎｔｒａ８ｘ８ＰｒｅｄＭｏｄｅおよびｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅを保持し、Ｉｎｔｒａ＿４ｘ４の場合は、Ｈ．２６４規格で定義されるＩｎｔｒａ４ｘ４ＰｒｅｄＭｏｄｅおよびｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅを保持し、それ以外のマクロブロックタイプの場合は情報を持たない。

【００１９】
・逆ＤＣＴ変換情報
逆ＤＣＴ変換に必要な、以下の情報を保持する。

【００２０】
－量子化係数
－ＤＣＴ係数
ＤＣＴ係数は、輝度成分、色差成分（ＣｂおよびＣｒ）がある。

【００２１】
図１において、画像復号部２００では、マクロブロック情報メモリ２１０に格納されているマクロブロックタイプによって、復号処理を分岐する。イントラマクロブロックである、Ｉｎｔｒａ＿１６ｘ１６、Ｉｎｔｒａ＿８ｘ８、Ｉｎｔｒａ＿４ｘ４は、画面内予測部２３０で画面内予測処理を行う。また、インターマクロブロックは動き補償部の、１６ｘ１６動き補償部２４０、１６ｘ８動き補償部２４１、８ｘ１６動き補償部２４２、８ｘ８動き補償部２４３、８ｘ４動き補償部２４４、４ｘ８動き補償部２４５、４ｘ４動き補償部２４６を使用し、マクロブロックタイプに応じた動き補償を行う。得られた予測画像と逆ＤＣＴ変換部２２０で得られた差分値を加算し、出力映像データ００２とするとともに、この復号された映像データをフレームバッファ２６０に格納する。

【００２２】
画像復号部２００におけるマクロブロックタイプ毎の詳細な処理を以下に示す。
【００２３】
・Ｉ＿ＰＣＭ
データ解析部で既に映像情報が得られているため、何も処理しない。
【００２４】
・Ｉｎｔｒａ＿１６ｘ１６
－輝度成分
（１）　Ｉｎｔｒａ１６ｘ１６ＰｒｅｄＭｏｄｅに従い、画面内予測処理を行う。
【００２５】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００２６】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００２７】
－色差成分
（１）　ｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅに従い、画面内予測処理を行う。
【００２８】
（２）　色差成分の逆ＤＣＴ変換を行う。
【００２９】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００３０】
・Ｉｎｔｒａ＿８ｘ８
－輝度成分
各サブブロックに対して、順に以下の処理を４回行う。
【００３１】
（１）　Ｉｎｔｒａ８ｘ８ＰｒｅｄＭｏｄｅに従い、画面内予測処理を行う。
【００３２】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００３３】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００３４】
－色差成分
（１）　ｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅに従い、画面内予測処理を行う。
【００３５】
（２）　色差成分の逆ＤＣＴ変換を行う。
【００３６】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００３７】
・Ｉｎｔｒａ＿４ｘ４
－輝度成分
各サブブロックに対して、順に以下の処理を１６回行う。
【００３８】

（１）　Ｉｎｔｒａ４ｘ４ＰｒｅｄＭｏｄｅに従い、画面内予測処理を行う。

【００３９】

（２）　輝度成分の逆ＤＣＴ変換を行う。

【００４０】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００４１】

－色差成分

（１）　ｉｎｔｒａ＿ｃｈｒｏｍａ＿ｐｒｅｄ＿ｍｏｄｅに従い、画面内予測処理を行う。

【００４２】

（２）　色差成分の逆ＤＣＴ変換を行う。

【００４３】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００４４】

・Ｉｎｔｅｒ＿１６ｘ１６

－輝度成分

（１）　画素単位を１６ｘ１６とし、輝度成分の動き補償処理を行う。

【００４５】

（２）　輝度成分の逆ＤＣＴ変換を行う。

【００４６】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００４７】

－色差成分

（１）　画素単位を８ｘ８とし、色差成分の動き補償処理を行う。

【００４８】

（２）　輝度成分の逆ＤＣＴ変換を行う。

【００４９】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００５０】

・Ｉｎｔｅｒ＿１６ｘ８

－輝度成分

（１）　画素単位を１６ｘ８とし、輝度成分の動き補償処理を２回行う。

【００５１】

（２）　輝度成分の逆ＤＣＴ変換を行う。

【００５２】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００５３】

－色差成分

（１）　画素単位を８ｘ４とし、色差成分の動き補償処理を２回行う。

【００５４】

（２）　輝度成分の逆ＤＣＴ変換を行う。

【００５５】

（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。

【００５６】
・Ｉｎｔｅｒ＿８ｘ１６
－輝度成分
（１） 画素単位を８ｘ１６とし、輝度成分の動き補償処理を２回行う。
【００５７】
（２） 輝度成分の逆ＤＣＴ変換を行う。
【００５８】
（３） 上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００５９】
－色差成分
（１） 画素単位を４ｘ８とし、色差成分の動き補償処理を２回行う。
【００６０】
（２） 輝度成分の逆ＤＣＴ変換を行う。
【００６１】
（３） 上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００６２】
・Ｉｎｔｅｒ＿８ｘ８
サブマクロブロックタイプに応じて、以降に示す画像復号処理を４回行う。
【００６３】
サブマクロブロックタイプ毎の処理は以下のように行う。
【００６４】
・Ｉｎｔｅｒ＿８ｘ８
－輝度成分
（１） 画素単位を８ｘ８とし、輝度成分の動き補償処理を行う。
【００６５】
（２） 輝度成分の逆ＤＣＴ変換を行う。
【００６６】
（３） 上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００６７】
－色差成分
（１） 画素単位を４ｘ４とし、色差成分の動き補償処理を行う。
【００６８】
（２） 輝度成分の逆ＤＣＴ変換を行う。
【００６９】
（３） 上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００７０】
・Ｉｎｔｅｒ＿８ｘ４
－輝度成分
（１） 画素単位を８ｘ４とし、輝度成分の動き補償処理を２回行う。
【００７１】
（２） 輝度成分の逆ＤＣＴ変換を行う。
【００７２】
（３） 上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００７３】
－色差成分

（１）　画素単位を４ｘ２とし、色差成分の動き補償処理を２回行う。
【００７４】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００７５】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００７６】
・Ｉｎｔｅｒ＿４ｘ８
－輝度成分
（１）　画素単位を４ｘ８とし、輝度成分の動き補償処理を２回行う。
【００７７】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００７８】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００７９】
－色差成分
（１）　画素単位を２ｘ４とし、色差成分の動き補償処理を２回行う。
【００８０】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００８１】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００８２】
・Ｉｎｔｅｒ＿４ｘ４
－輝度成分
（１）　画素単位を４ｘ４とし、輝度成分の動き補償処理を２回行う。
【００８３】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００８４】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００８５】
－色差成分
（１）　画素単位を２ｘ２とし、色差成分の動き補償処理を２回行う。
【００８６】
（２）　輝度成分の逆ＤＣＴ変換を行う。
【００８７】
（３）　上記（１）、（２）の結果を加算し、復号画像として結果をメモリに格納する。
【００８８】
ここで、輝度成分の動き補償処理および、色差成分の動き補償処理とは、以下の処理を指す。
【００８９】
・輝度成分の動き補償処理
－入力情報
・動き補償を行う画素単位
図３に示すブロックの分け方に基づいた、１６ｘ１６、１６ｘ８、８ｘ１６、８ｘ８、８ｘ４、４ｘ８、４ｘ４のいずれか。
【００９０】

・垂直方向のサブペル情報
モーションベクトルの垂直成分を４で割った余り。Ｈ．２６４規格におけるｘＦｒａｃに相当する。
【００９１】
・水平方向のサブペル情報
モーションベクトルの水平成分を４で割った余り。Ｈ．２６４規格におけるｘＦｒａｃに相当する。
【００９２】
・参照画素の左上の画素が格納されているメモリアドレス
前述のマクロブロック情報における、参照ピクチャインデックスおよび、モーションベクトルから算出される、Ｈ．２６４規格で規定の参照画素が格納されているメモリアドレス
・参照画素の１段下の画素までのメモリアドレス差分値
・出力先の左上の画素を格納するメモリアドレス
・出力先の画素の１段下の画素までのメモリアドレス差分値
以上の情報を基に、出力先のメモリに動き補償結果を格納する。動き補償処理の内部演算については、Ｈ．２６４規格で定義されているため、ここでの説明は割愛する。また、動き補償を行う画素単位毎に、速度最適化した専用の動き補償処理をＭＭＸ、または、ＳＳＥ２を用いて実装する必要があるが、その実装方法については、幾通りもあるため、ここでは、詳細な実装方法は割愛する。
【００９３】
動き補償処理を速度最適化する際の、基本的な考え方について、ＳＳＥ２で用いる１２８ｂｉｔ拡張レジスタを例に挙げて説明する。
【００９４】
ＳＳＥ２では、図４に示すように、１２８ｂｉｔのレジスタを連続した８個の１６ｂｉｔデータの集まりとして、加算、減算、乗算、シフト演算するＣＰＵ拡張命令セットを有している。この拡張命令セットを組み合わせて使用することにより、Ｈ．２６４規格における動き補償で必要な６タップフィルリングを、連続した８成分に対して同時に行うことができる。
【００９５】
この速度最適化を動き補償を行う画素単位毎に行い、２５６画素（１６ｘ１６画素）分を演算する速度を比較すると、以下の関係が得られる。
【００９６】
（高速）１６ｘ１６画素単位ｘ１回
１６ｘ８画素単位ｘ２回、８ｘ１６画素単位ｘ２回
８ｘ８画素単位ｘ４回
８ｘ４画素単位ｘ８回、４ｘ８画素単位ｘ８回
（低速）４ｘ４画素単位ｘ１６回
・色差成分の動き補償処理
－入力情報
・動き補償を行う画素単位
８ｘ８、８ｘ４、４ｘ８、４ｘ４、４ｘ２、２ｘ４、２ｘ２のいずれか
。
【００９７】
・垂直方向のサブペル情報
モーションベクトルの垂直成分を４で割った余り。Ｈ．２６４規格におけるｘＦｒａｃに相当する。
【００９８】
・水平方向のサブペル情報
モーションベクトルの水平成分を４で割った余り。Ｈ．２６４規格におけるｘＦｒａｃに相当する。

【００９９】

・参照画素の左上の画素が格納されているメモリアドレス

前述のマクロブロック情報における、参照ピクチャインデックスおよび、モーションベクトルから算出される、Ｈ．２６４規格で規定の参照画素が格納されているメモリアドレス

・参照画素の１段下の画素までのメモリアドレス差分値

・出力先の左上の画素を格納するメモリアドレス

・　出力先の画素の１段下の画素までのメモリアドレス差分値

以上の情報を基に、出力先のメモリに動き補償結果を格納する。動き補償処理の内部演算については、Ｈ．２６４規格で定義されているため、ここでの説明は割愛する。また、動き補償を行う画素単位毎に、速度最適化した専用の動き補償処理をＭＭＸ、または、ＳＳＥ２を用いて実装する必要があるが、その実装方法については、幾通りもあるため、ここでは割愛する。

【０１００】

ここからは、データ解析部におけるマクロブロック情報の決定方法について、その詳細を説明する。各マクロブロック情報は、Ｈ．２６４規格におけるｓｌｉｃｅ＿ｄａｔａ（）およびｍａｃｒｏｂｌｏｃｋ＿ｌａｙｅｒ（）のデータ列を、読み取ることで算出する。Ｈ．２６４規格で定義されるｍｂ＿ｔｙｐｅフィールドあるいは、Ｈ．２６４規格で定義されるｍｂ＿ｓｋｉｐ＿ｒｕｎフィールド（ＣＡＶＬＣのみ）または、ｍｂ＿ｓｋｉｐ＿ｆｌａｇフィールド（ＣＡＢＡＣのみ）を読み取り、Ｈ．２６４規格で定義されるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅを決定する。Ｈ．２６４規格で定義されるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅには、５６種類あり、Ｂ＿Ｄｉｒｅｃｔ＿１６ｘ１６、Ｂ＿Ｓｋｉｐの２種類を除いて、前述のマクロブロック情報におけるマクロブロックタイプを以下のように分類する。

【０１０１】

・マクロブロックタイプＩ＿ＰＣＭを割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ

Ｎａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ＝Ｉ＿ＰＣＭ

・マクロブロックタイプＩｎｔｒａ＿１６ｘ１６を割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ

Ｎａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ＝Ｉ＿１６ｘ１６＿０＿０＿０、Ｉ＿１６ｘ１６＿１＿０＿０、Ｉ＿１６ｘ１６＿２＿０＿０、Ｉ＿１６ｘ１６＿３＿０＿０、Ｉ＿１６ｘ１６＿０＿１＿０、Ｉ＿１６ｘ１６＿１＿１＿０、Ｉ＿１６ｘ１６＿２＿１＿０、Ｉ＿１６ｘ１６＿３＿１＿０、Ｉ＿１６ｘ１６＿０＿２＿０、Ｉ＿１６ｘ１６＿１＿２＿０、Ｉ＿１６ｘ１６＿２＿２＿０、Ｉ＿１６ｘ１６＿３＿２＿０、Ｉ＿１６ｘ１６＿０＿０＿１、Ｉ＿１６ｘ１６＿１＿０＿１、Ｉ＿１６ｘ１６＿２＿０＿１、Ｉ＿１６ｘ１６＿３＿０＿１、Ｉ＿１６ｘ１６＿０＿１＿１、Ｉ＿１６ｘ１６＿１＿１＿１、Ｉ＿１６ｘ１６＿２＿１＿１、Ｉ＿１６ｘ１６＿３＿１＿１、Ｉ＿１６ｘ１６＿０＿２＿１、Ｉ＿１６ｘ１６＿１＿２＿１、Ｉ＿１６ｘ１６＿２＿２＿１、Ｉ＿１６ｘ１６＿３＿２＿１

・マクロブロックタイプＩｎｔｒａ＿８ｘ８を割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ

Ｎａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ＝Ｉ＿ＮｘＮ　かつ　ｔｒａｎｓｆｏｒｍ＿ｓｉｚｅ＿８ｘ８＿ｆｌａｇフィールド＝１

・マクロブロックタイプＩｎｔｒａ＿４ｘ４を割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ

Ｎａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ＝Ｉ＿ＮｘＮ　かつ　ｔｒａｎｓｆｏｒｍ＿ｓｉｚｅ＿８ｘ８＿ｆｌａｇフィールド＝０

・マクロブロックタイプＩｎｔｅｒ＿１６ｘ１６を割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ

Ｎａｍｅ　ｏｆ　ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿１６ｘ１６、Ｐ＿Ｓｋｉｐ、Ｂ＿Ｌ０＿１６ｘ１６、Ｂ＿Ｌ１＿１６ｘ１６、Ｂ＿Ｂｉ＿１６ｘ１６

・マクロブロックタイプＩｎｔｅｒ＿１６ｘ８を割り当てるＮａｍｅ　ｏｆ　ｍｂ＿ｔ

ｙｐｅ
Ｎａｍｅ ｏｆ ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿Ｌ０＿１６ｘ８、Ｂ＿Ｌ０＿Ｌ０＿１６ｘ８、Ｂ＿Ｌ１＿Ｌ１＿１６ｘ８、Ｂ＿Ｌ０＿Ｌ１＿１６ｘ８、Ｂ＿Ｌ１＿Ｌ０＿１６ｘ８、Ｂ＿Ｌ０＿Ｂｉ＿１６ｘ８、Ｂ＿Ｌ１＿Ｂｉ＿１６ｘ８、Ｂ＿Ｂｉ＿Ｌ０＿１６ｘ８、Ｂ＿Ｂｉ＿Ｌ１＿１６ｘ８、Ｂ＿Ｂｉ＿Ｂｉ＿１６ｘ８
・マクロブロックタイプＩｎｔｅｒ＿８ｘ１６を割り当てるＮａｍｅ ｏｆ ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿Ｌ０＿８ｘ１６、Ｂ＿Ｌ０＿Ｌ０＿８ｘ１６、Ｂ＿Ｌ１＿Ｌ１＿８ｘ１６、Ｂ＿Ｌ０＿Ｌ１＿８ｘ１６、Ｂ＿Ｌ１＿Ｌ０＿８ｘ１６、Ｂ＿Ｌ０＿Ｂｉ＿８ｘ１６、Ｂ＿Ｌ１＿Ｂｉ＿８ｘ１６、Ｂ＿Ｂｉ＿Ｌ０＿８ｘ１６、Ｂ＿Ｂｉ＿Ｌ１＿８ｘ１６、Ｂ＿Ｂｉ＿Ｂｉ＿８ｘ１６
・マクロブロックタイプＩｎｔｅｒ＿８ｘ８を割り当てるＮａｍｅ ｏｆ ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｍｂ＿ｔｙｐｅ＝Ｐ＿８ｘ８、Ｐ＿８ｘ８ｒｅｆ０、Ｂ＿８ｘ８
マクロブロックタイプにＩｎｔｅｒ＿８ｘ８を割り当てたマクロブロックについては、さらにｓｕｂ＿ｍｂ＿ｔｙｐｅフィールドを読み取り、Ｎａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅがＢ＿Ｄｉｒｅｃｔ＿８ｘ８の場合を除いて、４つのサブマクロブロックタイプを以下のように割り当てる。

【０１０２】

・サブマクロブロックタイプＩｎｔｒａ＿８ｘ８を割り当てるＮａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿８ｘ８、Ｂ＿Ｌ０＿８ｘ８、Ｂ＿Ｌ１＿８ｘ８、Ｂ＿Ｂｉ＿８ｘ８
・サブマクロブロックタイプＩｎｔｒａ＿８ｘ４を割り当てるＮａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿８ｘ４、Ｂ＿Ｌ０＿８ｘ４、Ｂ＿Ｌ１＿８ｘ４、Ｂ＿Ｂｉ＿８ｘ４
・サブマクロブロックタイプＩｎｔｒａ＿４ｘ８を割り当てるＮａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿４ｘ８、Ｂ＿Ｌ０＿４ｘ８、Ｂ＿Ｌ１＿４ｘ８、Ｂ＿Ｂｉ＿４ｘ８
・サブマクロブロックタイプＩｎｔｒａ＿４ｘ４を割り当てるＮａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ
Ｎａｍｅ ｏｆ ｓｕｂ＿ｍｂ＿ｔｙｐｅ＝Ｐ＿Ｌ０＿４ｘ４、Ｂ＿Ｌ０＿４ｘ４、Ｂ＿Ｌ１＿４ｘ４、Ｂ＿Ｂｉ＿４ｘ４
以上のように、Ｂ＿Ｄｉｒｅｃｔ＿１６ｘ１６、Ｂ＿Ｓｋｉｐの２種類を除くインターマクロブロックにおいては、マクロブロックタイプおよびサブマクロブロックタイプは、Ｈ．２６４規格で定義されるＭｂＰａｒｔＷｉｄｔｈ、ＭｂＰａｒｔＨｅｉｇｈｔ、およびＳｕｂＭｂＰａｒｔＷｉｄｔｈ、ＳｕｂＭｂＰａｒｔＨｅｉｇｈｔと対になるように割り当てる。Ｂ＿Ｄｉｒｅｃｔ＿１６ｘ１６、Ｂ＿Ｓｋｉｐに対しては、以下の手順でマクロブロックタイプを割り当てる。
＜Ｂ＿Ｄｉｒｅｃｔ＿１６ｘ１６、Ｂ＿Ｓｋｉｐに対するマクロブロックタイプの割当手順＞
Ｈ．２６４規格で定義されるｓｌｉｃｅ＿ｈｅａｄｅｒ（）におけるｄｉｒｅｃｔ＿ｓｐａｔｉａｌ＿ｍｖ＿ｐｒｅｄ＿ｆｌａｇが１の場合、空間ダイレクトモーションベクトル解析処理を実行する。ｄｉｒｅｃｔ＿ｓｐａｔｉａｌ＿ｍｖ＿ｐｒｅｄ＿ｆｌａｇが０の場合、時間ダイレクトモーションベクトル解析処理を実行する。
＜空間ダイレクトモーションベクトル解析処理＞
図５に示すように、Ｈ．２６４規格で定義されるＭｉｎＰｏｓｉｔｉｖｅ処理を実行（

Ｓ００１０）し、０以上の最小となるｒｅｆ＿ｉｄｘを算出する。
ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０ ＝ ＭｉｎＰｏｓｉｔｉｖｅ（ ｒｅｆＩｄｘＬ０Ａ， ＭｉｎＰｏｓｉｔｉｖｅ（ ｒｅｆＩｄｘＬ０Ｂ， ｒｅｆＩｄｘＬ０Ｃ ） ）
ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１ ＝ ＭｉｎＰｏｓｉｔｉｖｅ（ ｒｅｆＩｄｘＬ１Ａ， ＭｉｎＰｏｓｉｔｉｖｅ（ ｒｅｆＩｄｘＬ１Ｂ， ｒｅｆＩｄｘＬ１Ｃ ） ）
ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０とｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１を確認して、条件に従い処理を実行する。
・ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０とｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１の値が共に０より小さい場合（Ｓ００１１でＹＥＳ）は、ゼロモーションベクトル設定を実行する（Ｓ０１００）。
・ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０とｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１の少なくとも一方の値が０の場合（Ｓ００１２でＹＥＳ）は、ＣｏｌＭＶ比較処理を実行する（Ｓ０２００）。
・上記以外の（ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０とｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１の値が共に０より大きい）場合（Ｓ００１２でＮＯ）は、中央値ベクトル設定を実行する（Ｓ０３００）。

【０１０３】

ゼロモーションベクトル設定（Ｓ０１００）は、図７に示すように、Ｓ０１０１の処理で
・ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０ ＝ ０
・ｍｖｐＬ０の水平成分 ＝ ０
・ｍｖｐＬ０の垂直成分 ＝ ０
・ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１ ＝ ０
・ｍｖｐＬ１の水平成分 ＝ ０
・ｍｖｐＬ１の垂直成分 ＝ ０
として、Ｉｎｔｅｒ＿１６ｘ１６設定を実行する（Ｓ１０００）。

【０１０４】

Ｉｎｔｅｒ＿１６ｘ１６設定（Ｓ１０００）は、図６に示すように、マクロブロックタイプをＩｎｔｒａ＿１６ｘ１６に設定（Ｓ１００１）し、予測ブロック番号０ の
・前方参照ピクチャインデックス＝ ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０
・前方参照モーションベクトルの水平成分＝ ｍｖｐＬ０の水平成分
・前方参照モーションベクトルの垂直成分＝ ｍｖｐＬ０の垂直成分
・後方参照ピクチャインデックス＝ ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１
・後方参照モーションベクトルの水平成分＝ ｍｖｐＬ１の水平成分
・後方参照モーションベクトルの垂直成分＝ ｍｖｐＬ１の垂直成分
に設定する（Ｓ１００２）。

【０１０５】

中央値ベクトル設定（Ｓ０３００）は、図８に示すように、ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０ が ０ より大きい場合（Ｓ０３０１でＹＥＳ）は、Ｈ．２６４規格で定義される前方の中央値ベクトルｍｖｐＬ０を算出（Ｓ０３０３）し、ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０ が ０ 以下の場合（Ｓ０３０１でＮＯ）は、ｍｖｐＬ０の水平成分と垂直成分を共に０とする（Ｓ０３０２）。

【０１０６】

また、後方ベクトルに対しても同様に、Ｓ０３０４、Ｓ０３０５、Ｓ０３０６を実行して、ｍｖｐＬ１を得る。そして、Ｉｎｔｅｒ＿１６ｘ１６設定を実行（Ｓ１０００）して、マクロブロックタイプおよびモーションベクトルを決定する。

【０１０７】

ＣｏｌＭＶ比較処理（Ｓ０２００）は、図９に示すように、ｍｉｎＰｏｓｉｔｉｖｅＲ

ｅｆＩｄｘＬ０ が ０ 以上の場合（Ｓ０２０１でＹＥＳ）は、Ｈ．２６４規格で定義される前方の中央値ベクトルｍｖｐＬ０を算出（Ｓ０２０３）し、ｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０ が ０ より小さい場合（Ｓ０２０１でＮＯ）は、ｍｖｐＬ０の水平成分と垂直成分を共に０とする（Ｓ０２０２）。また、後方ベクトルに対しても同様に、Ｓ０２０４、Ｓ０２０５、Ｓ０２０６を実行して、ｍｖｐＬ１を得る。ここで、算出したｍｖｐＬ０、ｍｖｐＬ１のモーションベクトルの成分が全て０の場合（Ｓ０２０７でＹＥＳ）、Ｉｎｔｅｒ＿１６ｘ１６設定を実行（Ｓ１０００）して、モーションベクトルを決定する。Ｓ０２０７でＮＯの場合は、ｃｏｌＭａｐを作成（Ｓ２０００）し、ｃｏｌＭａｐに基づきモーションベクトルを書き換え（Ｓ３０００）、ｃｏｌＭａｐに基づきマクロブロックタイプを決定（Ｓ４０００）する。

【０１０８】

ｃｏｌＭａｐの作成は、図１０に示すように、変数ｃｏｌＭａｐを初期化（Ｓ２００１）し、Ｈ．２６４規格で定義されるＣｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋ のマクロブロックタイプを確認する。Ｉｎｔｅｒ＿１６ｘ１６の場合（Ｓ２００２でＹＥＳ）は、ＣｏｌＭａｐ＿１６ｘ１６設定（Ｓ２１００）を行ない、Ｉｎｔｅｒ＿１６ｘ８の場合（Ｓ２００３でＹＥＳ）は、ＣｏｌＭａｐ＿１６ｘ８設定（Ｓ２２００）を行ない、Ｉｎｔｅｒ＿８ｘ１６の場合（Ｓ２００４でＹＥＳ）は、ＣｏｌＭａｐ＿８ｘ１６設定（Ｓ２３００）を行ない、Ｉｎｔｅｒ＿８ｘ８の場合（Ｓ２００５でＹＥＳ）は、各サブブロックに対してＣｏｌＭａｐ＿８ｘ８設定（Ｓ２４００）を行ない、上記以外の場合（Ｓ２００５でＮＯ）は、Ｃｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋ のマクロブロックタイプはイントラマクロブロックであるため何もせずに終了する。

【０１０９】

ＣｏｌＭａｐ＿１６ｘ１６設定は、図１１に示すように、Ｈ．２６４規格で定義されるＣｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋ の該当箇所のモーションベクトルｃｏｌＭＶおよび参照ピクチャインデックスｃｏｌＲｅｆＩｄｘを確認（Ｓ２１０１）し、ｃｏｌＲｅｆＩｄｘ が ０で、ｃｏｌＭＶの水平成分および垂直成分の絶対値が１以下の場合（Ｓ２１０１でＹＥＳ）、ｃｏｌＭａｐ を ０ｘＦＦＦＦ とする（Ｓ２１０２）。

【０１１０】

同様に、ＣｏｌＭａｐ＿１６ｘ８設定は図１２、ＣｏｌＭａｐ＿８ｘ１６設定は図１３に示すように、Ｃｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋの分割ブロック毎にモーションベクトルｃｏｌＭＶおよび参照ピクチャインデックスｃｏｌＲｅｆＩｄｘを確認し、ｃｏｌＭａｐの値を更新する。

【０１１１】

ＣｏｌＭａｐ＿８ｘ８作成は、図１４に示すように、Ｈ．２６４規格で定義されるＣｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋのサブマクロブロックタイプ毎に、Ｓ２４０２、Ｓ２４０３、Ｓ２４０４の判定で処理を分類し、ｃｏｌＳｕｂＭａｐを取得する（Ｓ２４１０、Ｓ２４２０、Ｓ２４３０、Ｓ２４４０）。得られたｃｏｌＳｕｂＭａｐは、指定されたブロックインデックスに応じてｃｏｌＭａｐに割り当てられる（Ｓ２４０５、Ｓ２４０６）。

【０１１２】

ＣｏｌＭａｐ＿８ｘ８設定は、図１５に、ＣｏｌＭａｐ＿８ｘ４設定は、図１６に、ＣｏｌＭａｐ＿４ｘ８設定は、図１６に、ＣｏｌＭａｐ＿４ｘ４設定は、図１７に示すように、Ｈ．２６４規格で定義されるＣｏ－ｌｏｃａｔｅｄ Ｍａｃｒｏｂｌｏｃｋ の該当箇所のモーションベクトルｃｏｌＭＶおよび参照ピクチャインデックスｃｏｌＲｅｆＩｄｘを確認し、ｃｏｌＲｅｆＩｄｘ が ０で、ｃｏｌＭＶの水平成分および垂直成分の絶対値が１以下の場合、ｃｏｌＳｕｂＭａｐ を 更新する。

【０１１３】

以上の手順によって得られたｃｏｌＭａｐの値は、算出しようとするマクロブロック内の空間ダイレクトモーションベクトルの内、どのアドレスのモーションベクトルが０になるかを示すものである。ｃｏｌＭａｐの各ビットは、図２０におけるモーションベクトル

のアドレスに対応しており、下位ビットがアドレス０に相当し、上位ビットがアドレス１５に相当する。この情報を基に、モーションベクトル書き換え処理、およびマクロブロックタイプ決定処理を行う。

【０１１４】

モーションベクトル書き換え処理は、図１９に示すように、図２０における全アドレスのモーションベクトルを前方はｍｖｐＬ０で、後方はｍｖｐＬ１に設定する（Ｓ３００１）。そして、全アドレスに対して、ｃｏｌＭａｐ　＆　（　０ｘ０００１　＜＜　ｉ　）が０より大きいかを判定し、かつ対象ブロックのｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ０が０の場合（Ｓ３００３でＹＥＳ）は、前方のモーションベクトルを０に書き換える（Ｓ３００４）。また、ｃｏｌＭａｐ　＆　（　０ｘ０００１　＜＜　ｉ　）が０より大きいかを判定し、かつ対象ブロックのｍｉｎＰｏｓｉｔｉｖｅＲｅｆＩｄｘＬ１が０の場合（Ｓ３００３でＹＥＳ）は、後方のモーションベクトルを０に書き換える（Ｓ３００４）。

マクロブロックタイプ決定処理は、図２１に示すように、得られたｃｏｌＭａｐの値によってマクロブロックタイプを決定する。例えば、Ｓ４００１では、ｃｏｌＭａｐが０ｘ００００または、０ｘＦＦＦＦであるかどうかを判定している。先に述べたように、ｃｏｌＭａｐは、モーションベクトルを０にするアドレスを示すものであり、ｃｏｌＭａｐが０ｘ００００の場合は、どのモーションベクトルも０にはならないことを意味する。また、０ｘＦＦＦＦは、すべて０になることを意味する。どちらの場合も、対象ブロック内のモーションベクトルは１種類であり、Ｉｎｔｅｒ＿１６ｘ１６とすることができる。他のマクロブロックタイプの決定手順も同様である。マクロブロックタイプにＩｎｔｅｒ＿８ｘ８を持つ場合は、図２２に示すサブブロックタイプ決定処理を行う。ここでもマクロブロックタイプの決定同様、ｃｏｌＭａｐの値によってサブブロックタイプを決定する。

以上のようにして得られた空間ダイレクトモーションベクトルは、最大で１６ｘ１６画素単位で動き補償処理を行うことが可能となる。

＜時間ダイレクトモーションベクトル解析処理＞

時間ダイレクトモーションベクトルの解析は、図２３に示すように、Ｈ．２６４規格で定義されるＣｏ－ｌｏｃａｔｅｄ　Ｍａｃｒｏｂｌｏｃｋのマクロブロック情報に格納されているマクロブロックタイプを算出対象のマクロブロックタイプとする（Ｓ５００１）。そして、対象のモーションベクトルを４ｘ４画素単位で算出し、算出対象のモーションベクトルとする。これにより得られる時間ダイレクトモーションベクトルは、最大で１６ｘ１６画素単位で動き補償処理を行うことが可能となる。

＜補足＞

上記実施の形態に基づいて、本発明に係る画像復号装置について説明してきたが本発明はこれに限定されるものではない。例えば、ｃｏｌＭａｐなどは、１６ｂｉｔのモーションベクトルの位置を表す変数として使用しているが、同様の結果が得られる別の変数を用いても良いし、判定条件で順不同にしても結果に変化がなければ、判定順序を入れ替えても良い。

【産業上の利用可能性】

【０１１５】

本発明に係る画像復号装置は、Ｈ．２６４の規格にしたがって圧縮された動画ストリームを復号する装置として活用することができる。

【図面の簡単な説明】

【０１１６】

【図１】本発明のＨ．２６４復号装置の構成図

【図２】マクロブロックデータ解析装置の構成図

【図３】動き補償処理のパターンを示す図

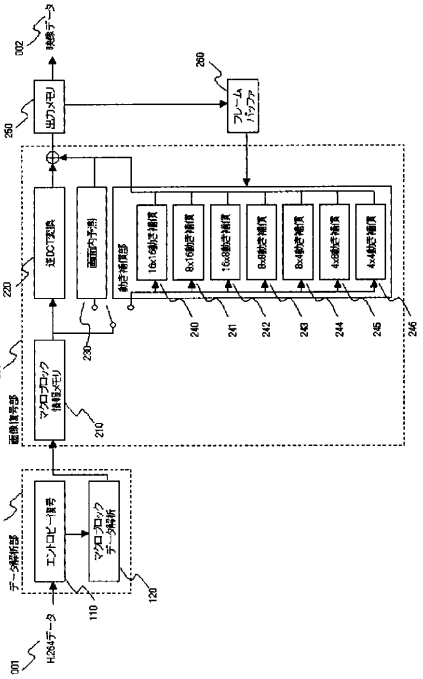
【図４】ＳＳＥ２で使用するレジスタの構成図

【図５】空間ダイレクトモーションベクトルの決定処理フローチャート

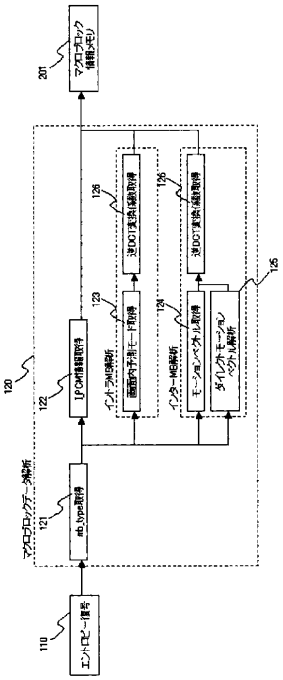
【図６】Ｉｎｔｅｒ＿１６ｘ１６設定処理フローチャート

【図７】ゼロモーションベクトル設定処理フローチャート
【図８】中央値ベクトル設定処理フローチャート
【図９】ＣｏｌＭＶ比較処理フローチャート
【図１０】ＣｏｌＭａｐ作成処理フローチャート
【図１１】ＣｏｌＭａｐ１６ｘ１６設定フローチャート
【図１２】ＣｏｌＭａｐ１６ｘ８設定フローチャート
【図１３】ＣｏｌＭａｐ８ｘ１６設定フローチャート
【図１４】ＣｏｌＭａｐ８ｘ８作成フローチャート
【図１５】ＣｏｌＭａｐ８ｘ８設定フローチャート
【図１６】ＣｏｌＭａｐ８ｘ４設定フローチャート
【図１７】ＣｏｌＭａｐ４ｘ８設定フローチャート
【図１８】ＣｏｌＭａｐ４ｘ４設定フローチャート
【図１９】モーションベクトル書き換え処理フローチャート
【図２０】モーションベクトルの配置を示す図
【図２１】マクロブロックタイプ決定処理フローチャート
【図２２】サブマクロブロックタイプ決定処理フローチャート
【図２３】時間ダイレクトモーションベクトル解析フローチャート
【符号の説明】
【０１１７】
００１　Ｈ．２６４符号データ
００２　映像データ
１００　データ解析部
１１０　エントロピー復号装置
１２０　マクロブロックデータ解析装置
１２１　ｍｂ＿ｔｙｐｅ取得装置
１２２　Ｉ＿ＰＣＭ情報取得装置
１２３　画面内予測モード取得装置
１２４　モーションベクトル取得装置
１２５　ダイレクトモーションベクトル解析装置
１２６　逆ＤＣＴ変換係数取得装置
２００　画像復号部
２１０　マクロブロック情報メモリ
２２０　逆ＤＣＴ変換装置
２３０　画面内予測装置
２４０　１６ｘ１６動き補償演算装置
２４１　１６ｘ８動き補償演算装置
２４２　８ｘ１６動き補償演算装置
２４３　８ｘ８動き補償演算装置
２４４　８ｘ４動き補償演算装置
２４５　４ｘ８動き補償演算装置
２４６　４ｘ４動き補償演算装置
２５０　出力メモリ
２６０　フレームバッファ

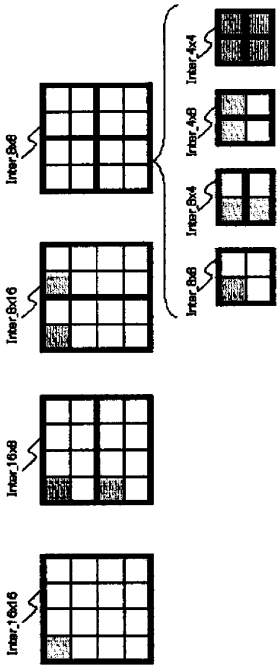
【図１】



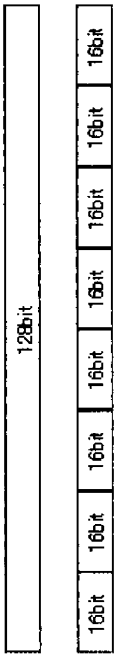
【図２】



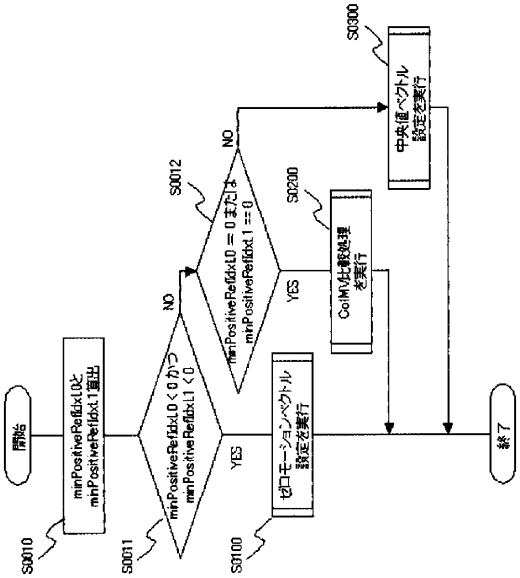
【図３】


Inter_8x8
Inter_8x16
Inter_16x8
Inter_16x16
Inter_4x4
Inter_4x8
Inter_8x4
Inter_8x8

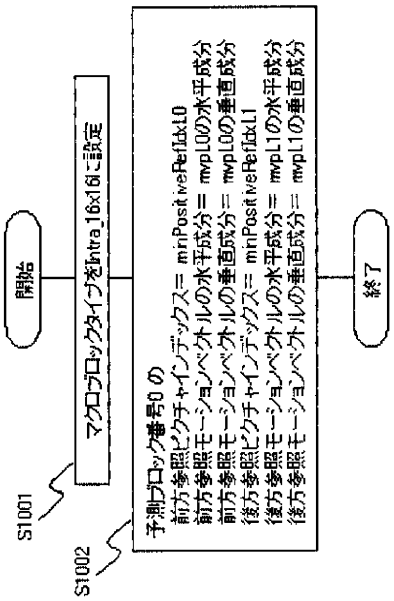
【図４】


128bit
16bit
16bit
16bit
16bit
16bit
16bit
16bit
16bit

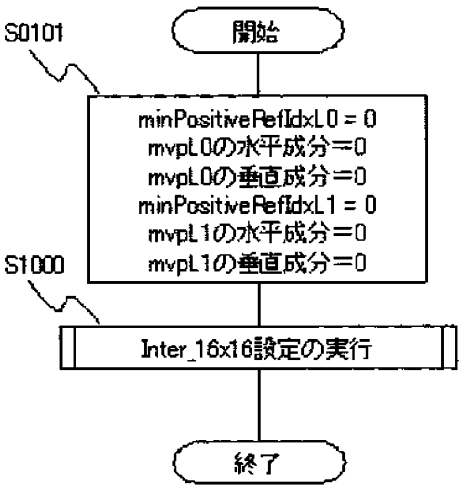
【図５】


開始
minPositiveRefIdxL0と
minPositiveRefIdxL1算出
S0010
minPositiveRefIdxL0 < 0かつ
minPositiveRefIdxL1 < 0
S0011
NO
YES
minPositiveRefIdxL0 = 0または
minPositiveRefIdxL1 == 0
S0012
NO
YES
中央値ベクトル
設定を実行
S0100
CoMV生成処理
を実行
S0200
中央値ベクトル
設定を実行
S0300
終了

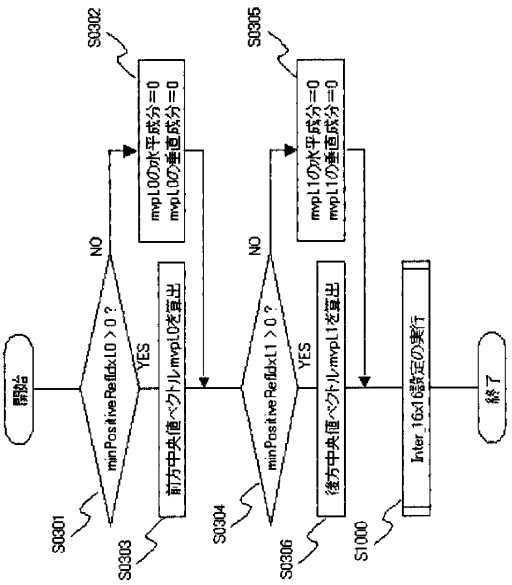
【図６】


開始
マクロブロックタイプをIntra_16x16に設定
S1001
予測ブロック番号0の
前方参照ピクチャインデックス = minPositiveRefIdxL0
前方参照モーションベクトルの水平成分 = mvpL0の水平成分
前方参照モーションベクトルの垂直成分 = mvpL0の垂直成分
後方参照ピクチャインデックス = minPositiveRefIdxL1
後方参照モーションベクトルの水平成分 = mvpL1の水平成分
後方参照モーションベクトルの垂直成分 = mvpL1の垂直成分
S1002
終了

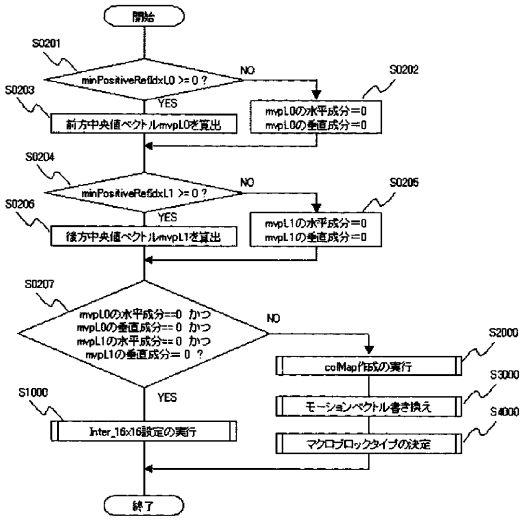
【図７】


S0101
開始
minPositiveRefIdxL0 = 0
mvpL0の水平成分＝0
mvpL0の垂直成分＝0
minPositiveRefIdxL1 = 0
mvpL1の水平成分＝0
mvpL1の垂直成分＝0
S1000
Inter_16x16設定の実行
終了

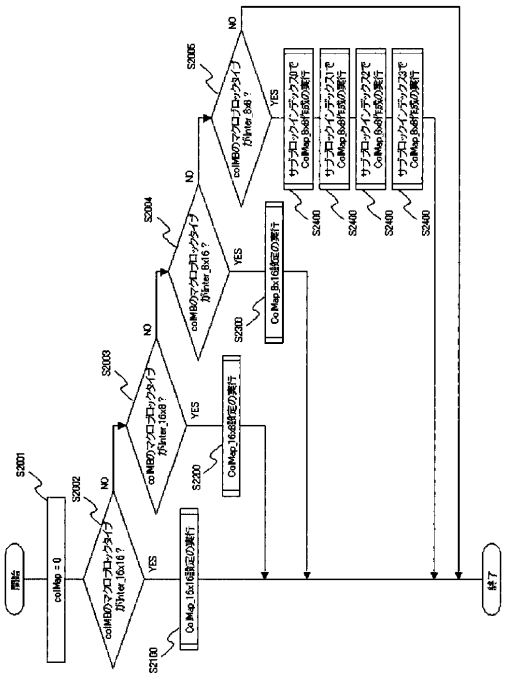
【図８】


開始
minPositiveRefIdxL0 > 0 ?
S0301
NO
YES
mvpL0の水平成分=0
mvpL0の垂直成分=0
S0302
前方中央値ベクトルmvpL0を算出
S0303
minPositiveRefIdxL1 > 0 ?
S0304
NO
YES
mvpL1の水平成分=0
mvpL1の垂直成分=0
S0305
後方中央値ベクトルmvpL1を算出
S0306
Inter_16x16設定の実行
S1000
終了

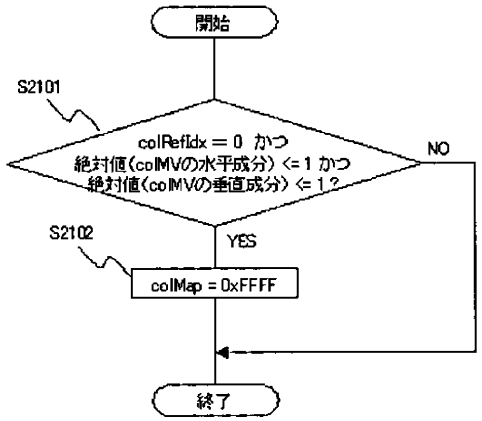
【図９】



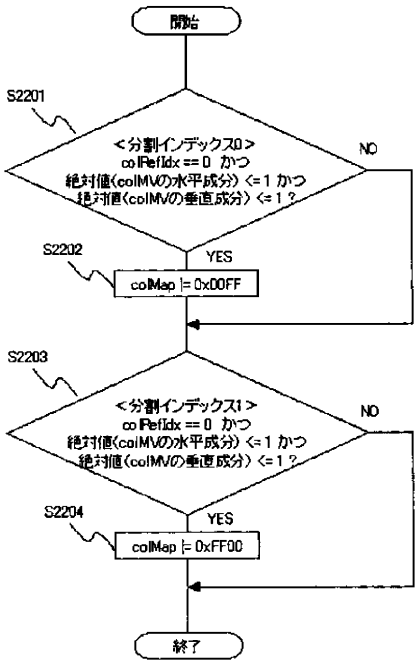
【図１０】



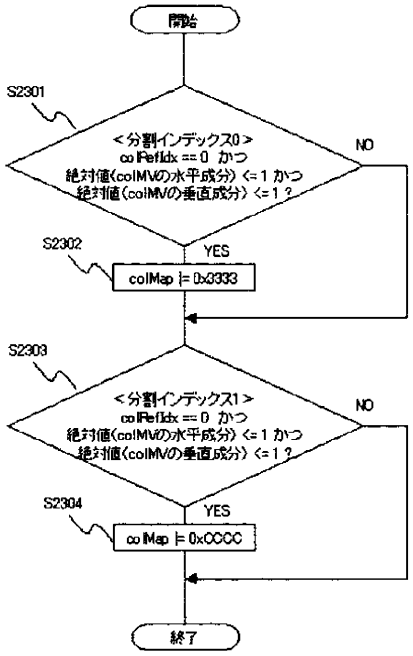
【図１１】



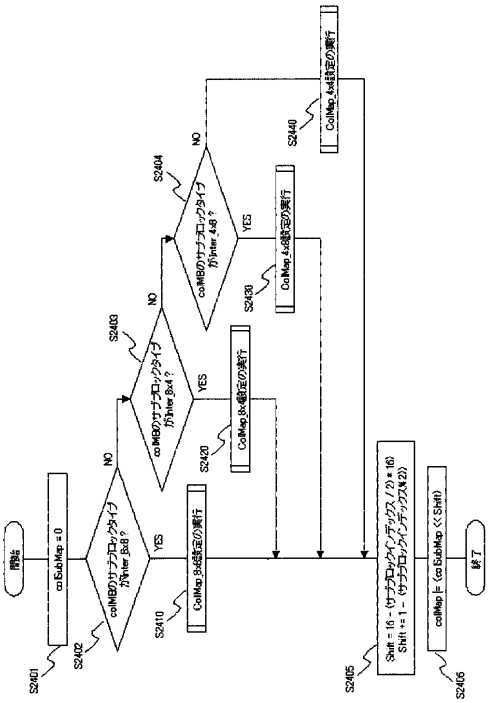
【図１２】



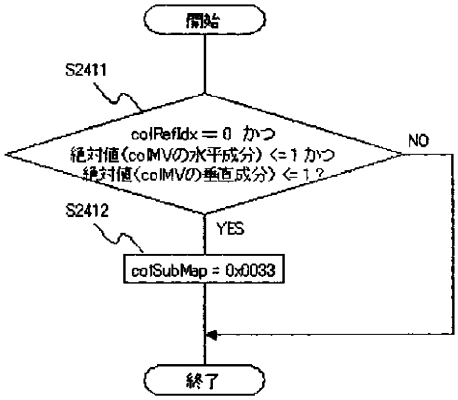
【図１３】



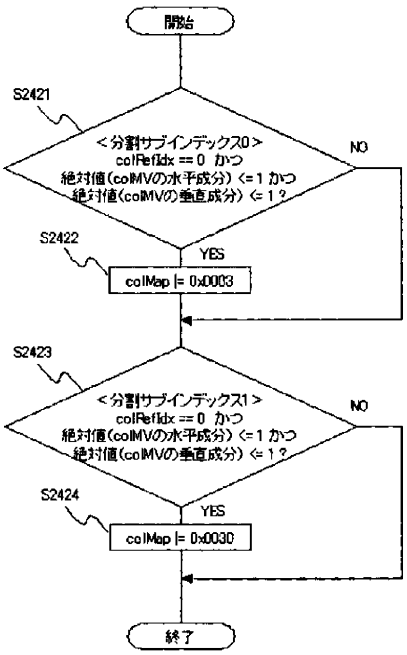
【図１４】



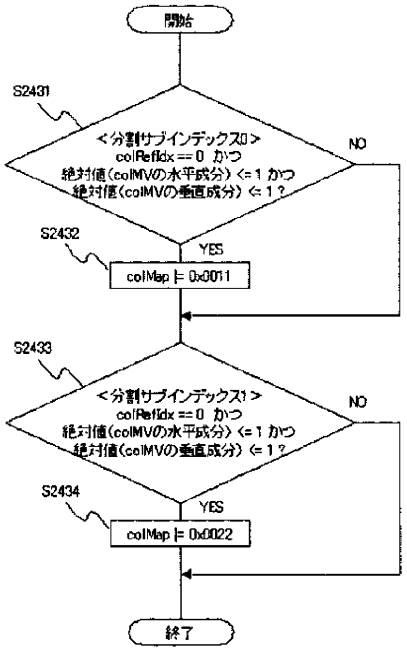
【図１５】



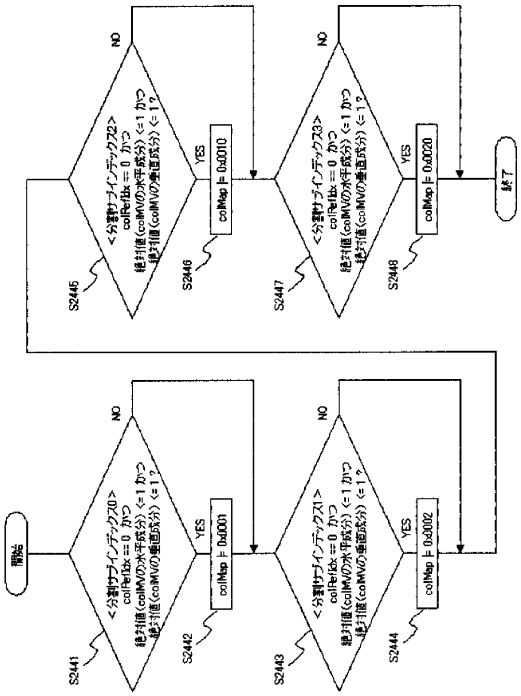
【図１６】



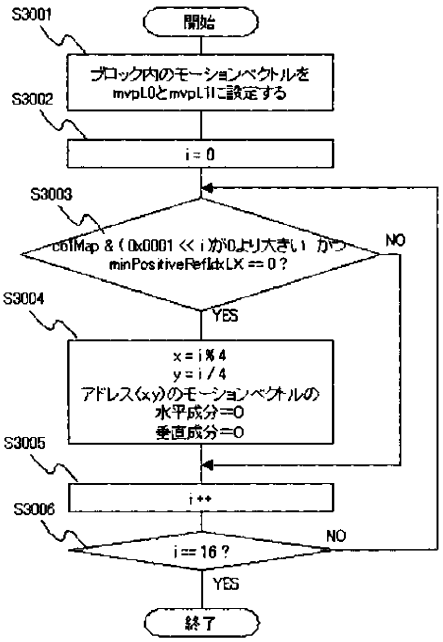
【図１７】



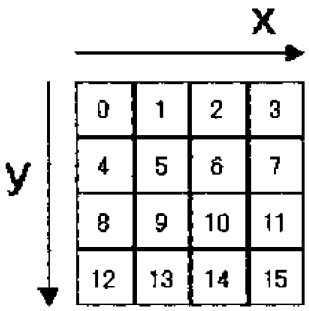
【図１８】



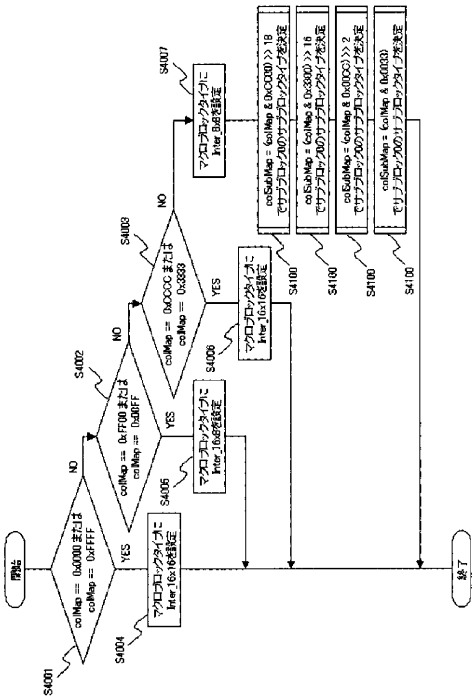
【図１９】



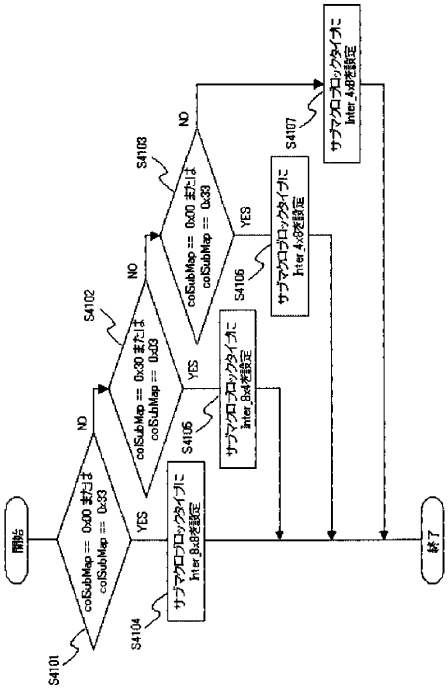
【図２０】



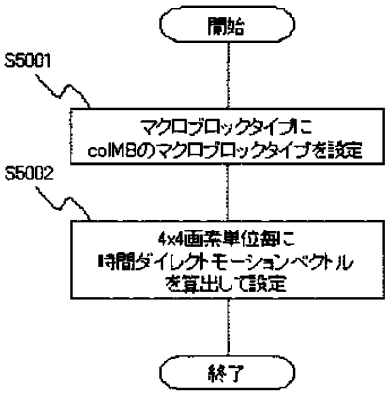
【図２１】


開始
S4001
S4002
S4003
S4004
S4005
S4006
S4007
S4100
終了

【図２２】


開始
S4101
S4102
S4103
S4104
S4105
S4106
S4107
終了

【図２３】


開始
S5001
マクロブロックタイプに
colMBのマクロブロックタイプを設定
S5002
4x4画素単位毎に
時間ダイレクトモーションベクトル
を算出して設定
終了