

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7637885号
(P7637885)

(45)発行日 令和7年3月3日(2025.3.3)

(24)登録日 令和7年2月20日(2025.2.20)

(51)国際特許分類

F I

H 0 4 N 19/52 (2014.01)

H 0 4 N 19/52

請求項の数 8 (全76頁)

(21)出願番号	特願2019-171786(P2019-171786)	(73)特許権者	513248094
(22)出願日	令和1年9月20日(2019.9.20)		合同会社 I P B r i d g e 1号
(65)公開番号	特開2020-150530(P2020-150530 A)		東京都千代田区神田神保町一丁目11番 地さくら総合事務所内
(43)公開日	令和2年9月17日(2020.9.17)	(74)代理人	100149548
審査請求日	令和4年9月9日(2022.9.9)		弁理士 松沼 泰史
(31)優先権主張番号	特願2018-247898(P2018-247898)	(74)代理人	100175824
(32)優先日	平成30年12月28日(2018.12.28)		弁理士 小林 淳一
(33)優先権主張国・地域又は機関		(74)代理人	100171446
	日本国(JP)		弁理士 高田 尚幸
(31)優先権主張番号	特願2019-42584(P2019-42584)	(72)発明者	中村 博哉
(32)優先日	平成31年3月8日(2019.3.8)		神奈川県横浜市神奈川区守屋町3丁目1 2番地
(33)優先権主張国・地域又は機関		(72)発明者	熊倉 徹
	日本国(JP)		神奈川県横浜市神奈川区守屋町3丁目1 最終頁に続く

(54)【発明の名称】 画像符号化装置、画像符号化方法及び画像符号化プログラム

(57)【特許請求の範囲】

【請求項1】

動画像をブロック単位でインター予測情報によるインター予測を用いて符号化する画像符号化装置であって、

符号化済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納部と、

符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出部と、

前記符号化対象ブロックとは異なるピクチャ内の前記符号化対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出部と、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出部と、
を備え、

前記履歴マージ候補導出部は、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像符号化装置。

【請求項 2】

動画像をブロック単位でインター予測情報によるインター予測を用いて符号化する画像符号化方法であって、

符号化済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記符号化対象ブロックとは異なるピクチャ内の前記符号化対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップと、
を備え、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像符号化方法。

【請求項 3】

動画像をブロック単位でインター予測情報によるインター予測を用いて符号化する画像符号化プログラムであって、

符号化済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記符号化対象ブロックとは異なるピクチャ内の前記符号化対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップとをコンピュータに実行させ、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像符号化プログラム。

【請求項 4】

動画像をブロック単位でインター予測を用いて符号化された符号化ビット列を復号する画像復号装置であって、

復号済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納部と、

復号対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出部と、

前記復号対象ブロックとは異なるピクチャ内の前記復号対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出部と、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出部と、

を備え、

10

20

30

40

50

前記履歴マージ候補導出部は、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像復号装置。

【請求項 5】

動画像をブロック単位でインター予測を用いて符号化された符号化ビット列を復号する画像復号方法であって、

復号済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

復号対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記復号対象ブロックとは異なるピクチャ内の前記復号対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップと、
を備え、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像復号方法。

【請求項 6】

動画像をブロック単位でインター予測を用いて符号化された符号化ビット列を復号する画像復号プログラムであって、

復号済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

復号対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記復号対象ブロックとは異なるピクチャ内の前記復号対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップとをコンピュータに実行させ、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする画像復号プログラム。

【請求項 7】

画像符号化装置で生成されたビットストリームを伝送する伝送方法であって、

符号化済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記符号化対象ブロックとは異なるピクチャ内の前記符号化対象ブロックと同一位置ま

10

20

30

40

50

たはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップと、

符号化情報を符号化しビットストリームを生成するビットストリーム生成ステップと、

前記ビットストリームを伝送するビットストリーム伝送ステップと、

を備え、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする伝送方法。

【請求項 8】

画像符号化装置で生成されたビットストリームを記録媒体に記録する記録方法であって、

符号化済ブロックのインター予測で用いたインター予測情報を履歴予測動きベクトル候補リストに先頭から順に格納する符号化情報格納ステップと、

符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間マージ候補を導出する空間マージ候補導出ステップと、

前記符号化対象ブロックとは異なるピクチャ内の前記符号化対象ブロックと同一位置またはその近傍に位置するブロックのインター予測情報から時間マージ候補を導出する時間マージ候補導出ステップと、

前記履歴予測動きベクトル候補リストに格納されたインター予測情報から履歴マージ候補を導出する履歴マージ候補導出ステップと、

符号化情報を符号化しビットストリームを生成するビットストリーム生成ステップと、

前記ビットストリームを前記記録媒体に記録するビットストリーム記録ステップと、

を備え、

前記履歴マージ候補導出ステップは、前記履歴予測動きベクトル候補リストに格納されたインター予測情報のうち、後ろから所定の数のインター予測情報について前記空間マージ候補のインター予測情報との比較を行い、インター予測情報の値が異なる場合に前記履歴マージ候補とし、後ろから前記所定の数より前方のインター予測情報について前記空間マージ候補のインター予測情報との比較を行わずに前記履歴マージ候補とする

ことを特徴とする記録方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像をブロックに分割して予測を行う画像符号化及び復号技術に関する。

【背景技術】

【0002】

画像の符号化及び復号では、処理の対象となる画像を所定数の画素の集合であるブロックに分割し、ブロック単位で処理をする。適切なブロックに分割し、画面内予測（イントラ予測）、画面間予測（インター予測）を適切に設定することにより、符号化効率が向上する。

【0003】

動画画像の符号化・復号では、符号化・復号済みのピクチャから予測するインター予測により符号化効率を向上している。特許文献 1 には、インター予測の際に、アフィン変換を適用する技術が記載されている。動画画像では、物体が拡大・縮小、回転といった変形を伴うことは珍しいことではなく、特許文献 1 の技術を適用することにより、効率的な符号化が可能となる。

【先行技術文献】

10

20

30

40

50

【特許文献】

【 0 0 0 4 】

【文献】特開平 9 - 1 7 2 6 4 4 号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 5 】

しかしながら、特許文献 1 の技術は画像の変換を伴うものであるため、処理負荷が多という課題がある。本発明は上記の課題に鑑み、低負荷で効率的な符号化技術を提供する。

【課題を解決するための手段】

【 0 0 0 6 】

上記課題を解決するために、本発明のある態様の画像符号化装置は、動画像をブロック単位でインター予測を用いて符号化する画像符号化装置であって、符号化済ブロックのインター予測情報を格納する符号化情報格納部と、符号化対象ブロックに空間的に近接するブロックのインター予測情報から空間インター予測情報候補を導出し、インター予測情報候補とする空間インター予測情報候補導出部と、前記符号化情報格納部に格納されたインター予測情報から履歴インター予測情報候補を導出し、インター予測情報候補とする履歴インター予測情報候補導出部と、を備え、前記履歴インター予測情報候補導出部は、前記符号化情報格納部に格納されている最新のインター予測情報から所定の数の要素について前記空間インター予測情報候補との比較を行い、すべてのインター予測情報が異なる場合にインター予測情報候補とする。

【発明の効果】

【 0 0 0 7 】

本発明によれば、高効率な画像符号化・復号処理を低負荷で実現することができる。

【図面の簡単な説明】

【 0 0 0 8 】

【図 1】本発明の実施の形態に係る画像符号化装置のブロック図である。

【図 2】本発明の実施の形態に係る画像復号装置のブロック図である。

【図 3】ツリーブロックを分割する動作を説明するためのフローチャートである。

【図 4】入力された画像をツリーブロックに分割する様子を示す図である。

【図 5】z - スキャンを説明する図である。

【図 6】ブロックの分割形状を示す図である。

【図 7】ブロックを 4 分割する動作を説明するためのフローチャートである。

【図 8】ブロックを 2 分割または 3 分割する動作を説明するためのフローチャートである。

【図 9】ブロック分割の形状を表現するためのシンタックスである。

【図 10】イントラ予測を説明するための図である。

【図 11】インター予測の参照ブロックを説明するための図である。

【図 12】符号化ブロック予測モードを表現するためのシンタックスである。

【図 13】インター予測に関するシンタックスエレメントとモードの対応を示す図である。

【図 14】制御点 2 点のアフィン変換動き補償を説明するための図である。

【図 15】制御点 3 点のアフィン変換動き補償を説明するための図である。

【図 16】図 1 のインター予測部 1 0 2 の詳細な構成のブロック図である。

【図 17】図 1 6 の通常予測動きベクトルモード導出部 3 0 1 の詳細な構成のブロック図である。

【図 18】図 1 6 の通常マージモード導出部 3 0 2 の詳細な構成のブロック図である。

【図 19】図 1 6 の通常予測動きベクトルモード導出部 3 0 1 の通常予測動きベクトルモード導出処理を説明するためのフローチャートである。

【図 20】通常予測動きベクトルモード導出処理の処理手順を表すフローチャートである。

【図 21】通常マージモード導出処理の手順を説明するフローチャートである。

【図 22】図 2 のインター予測部 2 0 3 の詳細な構成のブロック図である。

【図 23】図 2 2 の通常予測動きベクトルモード導出部 4 0 1 の詳細な構成のブロック図

10

20

30

40

50

である。

【図 2 4】図 2 2 の通常マージモード導出部 4 0 2 の詳細な構成のブロック図である。

【図 2 5】図 2 2 の通常予測動きベクトルモード導出部 4 0 1 の通常予測動きベクトルモード導出処理を説明するためのフローチャートである。

【図 2 6】図 1 6 のサブブロック予測動きベクトルモード導出部 3 0 3 の詳細な構成のブロック図である。

【図 2 7】図 2 2 のサブブロック予測動きベクトルモード導出部 4 0 3 の詳細な構成のブロック図である。

【図 2 8】図 1 6 のサブブロックマージモード導出部 3 0 4 の詳細な構成のブロック図である。

【図 2 9】図 2 2 のサブブロックマージモード導出部 4 0 4 の詳細な構成のブロック図である。

【図 3 0】アフィン継承予測動きベクトル候補導出を説明する図である。

【図 3 1】アフィン構築予測動きベクトル候補導出を説明する図である。

【図 3 2】アフィン継承マージ候補導出を説明する図である。

【図 3 3】アフィン構築マージ候補導出を説明する図である。

【図 3 4】アフィン継承予測動きベクトル候補導出のフローチャートである。

【図 3 5】アフィン構築予測動きベクトル候補導出のフローチャートである。

【図 3 6】アフィン継承マージ候補導出のフローチャートである。

【図 3 7】アフィン構築マージ候補導出のフローチャートである。

【図 3 8】履歴予測動きベクトル候補リスト初期化・更新処理手順を説明するフローチャートである。

【図 3 9】本発明の第 1 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における、同一要素確認処理手順のフローチャートである。

【図 4 0】履歴予測動きベクトル候補導出処理手順における、要素シフト処理手順のフローチャートである。

【図 4 1】履歴予測動きベクトル候補導出処理手順を説明するフローチャートである。

【図 4 2】本発明の第 1 の実施形態の履歴マージ候補導出処理手順を説明するフローチャートである。

【図 4 3】履歴予測動きベクトル候補リスト更新処理の一例を説明する図である。

【図 4 4】サブブロック時間マージ候補導出部 3 8 1 の動作を説明するためのフローチャートである。

【図 4 5】ブロックの隣接動き情報を導出する処理を説明するためのフローチャートである。

【図 4 6】テンポラル動きベクトルを導出する処理を説明するためのフローチャートである。

【図 4 7】インター予測情報の導出を説明するためのフローチャートである。

【図 4 8】サブブロック動き情報を導出する処理を説明するためのフローチャートである。

【図 4 9】ピクチャの時間的な前後関係を説明するための図である。

【図 5 0】通常予測動きベクトルモード導出部 3 0 1 における時間予測動きベクトル候補の導出処理を説明するためのフローチャートである。

【図 5 1】通常予測動きベクトルモード導出部 3 0 1 における時間予測動きベクトル候補の導出処理における、ColPicの導出処理を説明するためのフローチャートである。

【図 5 2】通常予測動きベクトルモード導出部 3 0 1 における時間予測動きベクトル候補の導出処理における、ColPicの符号化情報を導出する処理を説明するためのフローチャートである。

【図 5 3】インター予測情報の導出処理を説明するためのフローチャートである。

【図 5 4】符号化ブロック colCbのインター予測モードが双予測 (Pred_BI) のときの符号化ブロックのインター予測情報の導出処理手順を示すフローチャートである。

【図 5 5】動きベクトルのスケーリング演算処理手順を説明するためのフローチャートで

10

20

30

40

50

ある。

【図 5 6】時間マージ候補の導出処理を説明するためのフローチャートである。

【図 5 7】L 0 予測であって L 0 の参照ピクチャ (R e f L 0 P i c) が処理対象ピクチャ (C u r P i c) より前の時刻にある場合の動き補償予測を説明するための図である。

【図 5 8】L 0 予測であって L 0 予測の参照ピクチャが処理対象ピクチャより後の時刻にある場合の動き補償予測を説明するための図である。

【図 5 9】双予測であって L 0 予測の参照ピクチャが処理対象ピクチャより前の時刻にあって、L 1 予測の参照ピクチャが処理対象ピクチャより後の時刻にある場合の動き補償予測を説明するための図である。

【図 6 0】双予測であって L 0 予測の参照ピクチャと L 1 予測の参照ピクチャが処理対象ピクチャより前の時刻にある場合の動き補償予測の予測方向を説明するための図である。

10

【図 6 1】双予測であって L 0 予測の参照ピクチャと L 1 予測の参照ピクチャが処理対象ピクチャより後の時刻にある場合の動き補償予測の予測方向を説明するための図である。

【図 6 2】平均マージ候補導出処理手順を説明するフローチャートである。

【図 6 3】マージ差分動きベクトルに関する情報を示す表である。

【図 6 4】マージ差分動きベクトルの導出を説明する図である。

【図 6 5】履歴予測動きベクトル候補リストの確認の際に、履歴予測動きベクトル候補リストの後ろからオフセット値 hMvpldxOffset で指定した個数の要素を説明する図である。

【図 6 6】本発明の第 2 の実施形態の履歴マージ候補導出処理手順を説明するフローチャートである。

20

【図 6 7】本発明の第 3 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における、同一要素確認処理手順のフローチャートである。

【図 6 8】本発明の第 4 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における、同一要素確認処理手順のフローチャートである。

【図 6 9】本発明の実施の形態の符号化復号装置のハードウェア構成の一例を説明するための図である。

【発明を実施するための形態】

【 0 0 0 9 】

本実施の形態において使用する技術、及び技術用語を定義する。

【 0 0 1 0 】

30

< ツリーブロック >

実施の形態では、所定の大きさを符号化・復号処理対象画像を均等分割する。この単位をツリーブロックと定義する。図 4 に示す通り、本実施の形態では、ツリーブロックのサイズを 1 2 8 × 1 2 8 画素と設定するが、ツリーブロックのサイズはこれに限定されるものではなく、任意のサイズを設定してよい。処理対象 (符号化処理においては符号化対象、復号処理においては復号対象に対応する。) のツリーブロックは、ラスタスキャン順、すなわち左から右、上から下の順序で切り替わる。各ツリーブロックの内部は、さらに再帰的な分割が可能である。ツリーブロック分割後の、符号化・復号の対象となるブロックを符号化ブロックと定義する。また、ツリーブロック、符号化ブロックを総称してブロックと定義する。適切なブロック分割を行うことにより効率的な符号化が可能となる。ツリーブロックのサイズは、符号化装置と復号装置で予め取り決めた固定値とすることもできるし、符号化装置が決定したツリーブロックのサイズを復号装置に伝送するような構成をとることもできる。

40

【 0 0 1 1 】

< 予測モード >

処理対象符号化ブロック単位で、処理対象画像の処理済み (符号化処理においては符号化が完了した信号を復号した画像、画像信号、ツリーブロック、ブロック、符号化ブロック等に用い、復号処理においては復号が完了した画像、画像信号、ツリーブロック、ブロック、符号化ブロック等に用いる。) の周囲の画像信号から予測を行うイントラ予測 (M O D E _ I N T R A) 、及び処理済み画像の画像信号から予測を行うインター予測 (M O D E _ I N T

50

ER) を切り替える。このイントラ予測 (MODE_INTRA) とインター予測 (MODE_INTER) を識別するモードを予測モード (PredMode) と定義する。予測モード (PredMode) はイントラ予測 (MODE_INTRA)、またはインター予測 (MODE_INTER) を値として持つ。

【0012】

< インター予測 >

処理済み画像の画像信号から予測を行うインター予測では、複数の処理済み画像を参照ピクチャとして用いることができる。複数の参照ピクチャを管理するため、L0 (参照リスト0) とL1 (参照リスト1) の2種類の参照リストを定義し、それぞれ参照インデックスを用いて参照ピクチャを特定する。PスライスではL0予測 (Pred_L0) が利用可能である。BスライスではL0予測 (Pred_L0)、L1予測 (Pred_L1)、双予測 (Pred_BI) が利用可能である。L0予測 (Pred_L0) はL0で管理されている参照ピクチャを参照するインター予測であり、L1予測 (Pred_L1) はL1で管理されている参照ピクチャを参照するインター予測である。双予測 (Pred_BI) はL0予測とL1予測が共に行われ、L0とL1のそれぞれで管理されている1つずつの参照ピクチャを参照するインター予測である。L0予測、L1予測、双予測を特定する情報を、インター予測モードと定義する。以降の処理において出力に添え字LXが付いている定数、変数に関しては、L0、L1ごとに処理が行われることを前提とする。

【0013】

< 予測動きベクトルモード >

予測動きベクトルモードは、予測動きベクトルを特定するためのインデックス、差分動きベクトル、インター予測モード、参照インデックスを伝送し、処理対象ブロックのインター予測情報を決定するモードである。予測動きベクトルは、処理対象ブロックに隣接する処理済みブロック、または処理済み画像に属するブロックで処理対象ブロックと同一位置またはその付近 (近傍) に位置するブロックから導出した予測動きベクトル候補と、予測動きベクトルを特定するためのインデックスから導出する。

【0014】

< マージモード >

マージモードは、差分動きベクトル、参照インデックスを伝送せずに、処理対象ブロックに隣接する処理済みブロック、または処理済み画像に属するブロックで処理対象ブロックと同一位置またはその付近 (近傍) に位置するブロックのインター予測情報から、処理対象ブロックのインター予測情報を導出するモードである。

【0015】

処理対象ブロックに隣接する処理済みブロック、およびその処理済みブロックのインター予測情報を空間マージ候補と定義する。処理済み画像に属するブロックで処理対象ブロックと同一位置またはその付近 (近傍) に位置するブロック、およびそのブロックのインター予測情報から導出されるインター予測情報を時間マージ候補と定義する。各マージ候補はマージ候補リストに登録され、マージインデックスにより処理対象ブロックの予測で使用するマージ候補を特定する。

【0016】

< 隣接ブロック >

図11は、予測動きベクトルモード、マージモードで、インター予測情報を導出するために参照する参照ブロックを説明する図である。A0, A1, A2, B0, B1, B2, B3は、処理対象ブロックに隣接する処理済みブロックである。T0は、処理済み画像に属するブロックで、処理対象画像の処理対象符号化ブロックと同一位置またはその付近 (近傍) に位置するブロックである。

【0017】

A1, A2は、処理対象符号化ブロックの左側に位置し、処理対象符号化ブロックに隣接するブロックである。B1, B3は、処理対象符号化ブロックの上側に位置し、処理対象符号化ブロックに隣接するブロックである。A0, B0, B2はそれぞれ、処理対象符

10

20

30

40

50

号化ブロックの左下、右上、左上に位置するブロックである。

【 0 0 1 8 】

予測動きベクトルモード、マージモードにおいて隣接ブロックをどのように扱うかの詳細については後述する。

【 0 0 1 9 】

< アフィン変換動き補償 >

アフィン変換動き補償は、符号化ブロックを所定単位のサブブロックに分割し、各サブブロックに対して個別に動きベクトルを設定して動き補償を行うものである。各サブブロックの動きベクトルは、処理対象ブロックに隣接する処理済みブロック、または処理済み画像に属するブロックで処理対象ブロックと同一位置またはその付近（近傍）に位置するブロックのインター予測情報から導出する1つ以上の制御点に基づき導出する。本実施の形態では、サブブロックのサイズを4×4画素とするが、サブブロックのサイズはこれに限定されるものではないし、画素単位で動きベクトルを導出してよい。

【 0 0 2 0 】

図14に、制御点が2つの場合のアフィン変換動き補償の例を示す。この場合、2つの制御点が水平方向成分、垂直方向成分の2つのパラメータを有するため、制御点が2つの場合のアフィン変換を、4パラメータアフィン変換と呼称する。図14のCP1、CP2が制御点である。図15に、制御点が3つの場合のアフィン変換動き補償の例を示す。この場合、3つの制御点が水平方向成分、垂直方向成分の2つのパラメータを有するため、制御点が3つの場合のアフィン変換を、6パラメータアフィン変換と呼称する。図15のCP1、CP2、CP3が制御点である。

【 0 0 2 1 】

アフィン変換動き補償は、予測動きベクトルモードおよびマージモードのいずれのモードにおいても利用可能である。予測動きベクトルモードでアフィン変換動き補償を適用するモードをサブブロック予測動きベクトルモードと定義し、マージモードでアフィン変換動き補償を適用するモードをサブブロックマージモードと定義する。

【 0 0 2 2 】

< インター予測のシンタックス >

図12、図13を用いて、インター予測に関するシンタックスを説明する。図12のmerge_flagは、処理対象符号化ブロックをマージモードとするか、予測動きベクトルモードとするかを示すフラグである。merge_affine_flagは、マージモードの処理対象符号化ブロックでサブブロックマージモードを適用するか否かを示すフラグである。inter_affine_flagは、予測動きベクトルモードの処理対象符号化ブロックでサブブロック予測動きベクトルモードを適用するか否かを示すフラグである。cu_affine_type_flagは、サブブロック予測動きベクトルモードにおいて、制御点の数を決定するためのフラグである。図13に各シンタックスエレメントの値と、それに対応する予測方法を示す。merge_flag=1, merge_affine_flag=0は、サブブロックマージでないマージモードである、通常マージモードに対応する。merge_flag=1, merge_affine_flag=1は、サブブロックマージモードに対応する。merge_flag=0, inter_affine_flag=0は、サブブロック予測動きベクトルモードでない予測動きベクトルマージである、通常予測動きベクトルモードに対応する。merge_flag=0, inter_affine_flag=1は、サブブロック予測動きベクトルモードに対応する。merge_flag=0, inter_affine_flag=1の場合は、さらにcu_affine_type_flagを伝送し、制御点の数を決定する。

【 0 0 2 3 】

< P O C >

P O C (Picture Order Count) は符号化されるピクチャに関連付けられる変数とし、ピクチャの出力順序で1ずつ増加する値が設定される。P O Cの値によって、同じピクチャであるかを判別したり、出力順序でのピクチャ間の前後関係を判別したり、ピクチャ間の距離を導出したりすることができる。例えば、2つのピクチャのP O Cが同じ値を持つ場合、同一のピクチャであると判断できる。2つのピクチャのP O Cが違う値を持つ場合

10

20

30

40

50

、POCの値が小さいピクチャのほうが、先に出力されるピクチャであると判断でき、2つのピクチャのPOCの差が時間軸方向でのピクチャ間距離を示す。

【0024】

(第1の実施の形態)

本発明の第1の実施の形態に係る画像符号化装置100及び画像復号装置200について説明する。

【0025】

図1は、第1の実施の形態に係る画像符号化装置100のブロック図である。実施の形態の動画像符号化装置は、画像符号化装置100、ブロック分割部101、インター予測部102、イントラ予測部103、復号画像メモリ104、予測方法決定部105、残差信号生成部106、直交変換・量子化部107、ビット列符号化部108、逆量子化・逆直交変換部109、復号画像信号重畳部110、および符号化情報格納メモリ111を備える。

10

【0026】

ブロック分割部101は、入力した画像を再帰的に分割して、符号化ブロックを生成する。ブロック分割部101は、分割対象となるブロックを水平方向と垂直方向にそれぞれ分割する4分割部と、分割対象となるブロックを水平方向または垂直方向のいずれかに分割する2-3分割部を含む。生成した処理対象符号化ブロックの画像信号を、インター予測部102、イントラ予測部103および残差信号生成部106に供給する。また、決定した再帰分割構造を示す情報をビット列符号化部108に供給する。ブロック分割部101の詳細な動作は後述する。

20

【0027】

インター予測部102は、処理対象符号化ブロックのインター予測を行う。符号化情報格納メモリに格納されているインター予測情報、復号画像メモリ104に格納されている復号済みの画像信号から複数のインター予測情報の候補を導出し、複数の候補の中から適したインター予測モードを選択し、選択されたインター予測モード、及び選択されたインター予測モードに応じた予測画像信号を予測方法決定部105に供給する。インター予測部102の詳細な構成と動作は後述する。

【0028】

イントラ予測部103は、処理対象符号化ブロックのイントラ予測を行う。復号画像メモリ104に格納されている復号済みの画像信号からイントラ予測により予測画像信号を生成し、複数のイントラ予測モードの中から適したイントラ予測モードを選択し、選択されたイントラ予測モード、及び選択されたイントラ予測モードに応じた予測画像信号を予測方法決定部105に供給する。図10にイントラ予測の例を示す。図10(a)は、イントラ予測の予測方向とイントラ予測モード番号の対応を示したものである。例えば、イントラ予測モード50は、垂直方向に画素をコピーすることによりイントラ予測画像を生成する。イントラ予測モード1は、DCモードであり、処理対象ブロックのすべての画素値を参照画素の平均値とするモードである。イントラ予測モード0はPlanarモードであり、垂直方向・水平方向の参照画素から2次元的なイントラ予測画像を作成するモードである。図10(b)は、イントラ予測モード40の場合のイントラ予測画像を生成する例である。処理対象ブロックの各画素に対し、イントラ予測モードの示す方向の参照画素の値をコピーする。イントラ予測モードの参照画素が整数位置でない場合には、周辺の整数位置の参照画素値から補間により参照画素値を決定する。

30

40

【0029】

復号画像メモリ104は、復号画像信号重畳部110で生成した復号画像を格納する。復号画像メモリに格納されている復号画像は、インター予測部102、イントラ予測部103に供給する。

【0030】

予測方法決定部105は、各予測に対して、符号化情報及び残差信号の符号量、予測画像信号と処理対象画像信号との間の歪量等を用いて評価することにより、最適な予測モー

50

ド（インター予測またはイントラ予測）を決定する。インター予測のマージモードの場合は、マージインデックス、サブブロックマージモードか否かを示す情報（サブブロックマージフラグ）の符号化情報をビット列符号化部 108 に供給し、インター予測の予測動きベクトルモードの場合はインター予測モード、予測動きベクトルインデックス、L0、L1の参照インデックス、差分動きベクトル、サブブロックモードか否かを示す情報（サブブロック予測動きベクトルフラグ）等の符号化情報をビット列符号化部 108 に供給する。決定した符号化情報を符号化情報格納メモリ 111 に供給する。

【0031】

残差信号生成部 106 は、処理対象の画像信号から予測画像信号を減ずることにより残差信号を生成し、直交変換・量子化部 107 に供給する。

【0032】

直交変換・量子化部 107 は、残差信号に対して量子化パラメータに応じて直交変換及び量子化を行い直交変換・量子化された残差信号を生成し、ビット列符号化部 108 と逆量子化・逆直交変換部 109 に供給する。

【0033】

ビット列符号化部 108 は、シーケンス、ピクチャ、スライス、符号化ブロック単位の情報に加えて、符号化ブロック毎に予測方法決定部 105 によって決定された予測方法に応じた符号化情報を符号化する。具体的には、符号化ブロック毎の予測モードPredMode、分割モードPartMode、インター予測（PRED_INTER）の場合、マージモードか否かを判別するフラグ、サブブロックマージフラグ、マージモードの場合はマージインデックス、マージモードでない場合はインター予測モード、予測動きベクトルインデックス、差分動きベクトルに関する情報、サブブロック予測動きベクトルフラグ等の符号化情報を後述する規定のシンタックス規則に従って符号化し、第1の符号化ビット列を生成する。また、ビット列符号化部 108 は、直交変換及び量子化された残差信号を規定のシンタックス規則に従ってエントロピー符号化して第2の符号化ビット列を生成する。第1の符号化ビット列と第2の符号化ビット列を規定のシンタックス規則に従って多重化し、ビットストリームを出力する。

【0034】

逆量子化・逆直交変換部 109 は、直交変換・量子化部 107 から供給された直交変換・量子化された残差信号を逆量子化及び逆直交変換して残差信号を算出し、復号画像信号重畳部 110 に供給する。

【0035】

復号画像信号重畳部 110 は、予測方法決定部 105 による決定に応じた予測画像信号と逆量子化・逆直交変換部 109 で逆量子化及び逆直交変換された残差信号を重畳して復号画像を生成し、復号画像メモリ 104 に格納する。なお、復号画像に対して符号化によるブロック歪等の歪を減少させるフィルタリング処理を施した後、復号画像メモリ 104 に格納してもよい。

【0036】

符号化情報格納メモリ 111 は、予測方法決定部 105 で決定した、予測モード（インター予測またはイントラ予測）等の符号化情報を格納する。符号化情報格納メモリ 111 が格納する符号化情報は、インター予測の場合は、決定した動きベクトル、参照リスト、参照インデックスに加え、インター予測のマージモードの場合は、マージインデックス、サブブロックマージモードか否かを示す情報（サブブロックマージフラグ）の符号化情報、インター予測の予測動きベクトルモードの場合はインター予測モード、予測動きベクトルインデックス、L0、L1の参照インデックス、差分動きベクトル、サブブロックモードか否かを示す情報（サブブロック予測動きベクトルフラグ）、イントラ予測の場合は、決定したイントラ予測モード等である。符号化情報格納メモリ 111 で管理される履歴候補リストの構築については後述する。

【0037】

図2は図1の動画像符号化装置に対応した本発明の実施の形態に係る動画像復号装置の

10

20

30

40

50

構成を示すブロックである。実施の形態の動画像復号装置は、ビット列復号部 201、ブロック分割部 202、インター予測部 203、イントラ予測部 204、符号化情報格納メモリ 205、逆量子化・逆直交変換部 206、復号画像信号重畳部 207、および復号画像メモリ 208 を備える。

【0038】

図 2 の動画像復号装置の復号処理は、図 1 の動画像符号化装置の内部に設けられている復号処理に対応するものであるから、図 2 の符号化情報格納メモリ 205、逆量子化・逆直交変換部 206、復号画像信号重畳部 207、および復号画像メモリ 208 の各構成は、図 1 の動画像符号化装置の逆量子化・逆直交変換部 109、復号画像信号重畳部 110、符号化情報格納メモリ 111、および復号画像メモリ 104 の各構成とそれぞれ対応する機能を有する。

10

【0039】

ビット列復号部 201 に供給されるビットストリームは規定のシンタックスの規則に従って分離する。分離された第 1 の符号化ビット列を復号し、シーケンス、ピクチャ、スライス、符号化ブロック単位の情報、及び、符号化ブロック単位の符号化情報を得る。具体的には、符号化ブロック単位でインター予測 (PRED_INTER) かイントラ予測 (PRED_INTRA) かを判別する予測モード PredMode、分割モード PartMode、インター予測 (PRED_INTER) の場合、マージモードか否かを判別するフラグ、マージモードの場合はマージインデックス、サブブロックマージフラグ、予測動きベクトルモードである場合はインター予測モード、予測動きベクトルインデックス、差分動きベクトル、サブブロック予測動きベクトルフラグ等に関する符号化情報を後述する規定のシンタックス規則に従って復号し、符号化情報をインター予測部 203 またはイントラ予測部 204、および符号化情報格納メモリ 205 に供給する。分離した第 2 の符号化ビット列を復号して直交変換・量子化された残差信号を算出し、直交変換・量子化された残差信号を逆量子化・逆直交変換部 206 に供給する。

20

【0040】

インター予測部 203 は、処理対象の符号化ブロックの予測モード PredMode がインター予測 (PRED_INTER) で予測動きベクトルモードである時に、符号化情報格納メモリ 205 に記憶されている既に復号された画像信号の符号化情報を用いて、複数の予測動きベクトルの候補を導出して後述する予測動きベクトル候補リストに登録し、予測動きベクトル候補リストに登録された複数の予測動きベクトルの候補の中から、ビット列復号部 201 で復号され供給される予測動きベクトルインデックスに応じた予測動きベクトルを選択し、ビット列復号部 201 で復号された差分ベクトルと選択された予測動きベクトルから動きベクトルを算出し、他の符号化情報とともに符号化情報格納メモリ 205 に格納する。ここで供給・格納する符号化ブロックの符号化情報は、予測モード PredMode、分割モード PartMode、L0 予測、及び L1 予測を利用するか否かを示すフラグ predFlagL0[xP][yP]、predFlagL1[xP][yP]、L0、L1 の参照インデックス refIdxL0[xP][yP]、refIdxL1[xP][yP]、L0、L1 の動きベクトル mvL0[xP][yP]、mvL1[xP][yP] 等である。ここで、xP、yP はピクチャ内での符号化ブロックの左上の画素の位置を示すインデックスである。予測モード PredMode がインター予測 (MODE_INTER) で、インター予測モードが L0 予測 (Pred_L0) の場合、L0 予測を利用するか否かを示すフラグ predFlagL0 は 1、L1 予測を利用するか否かを示すフラグ predFlagL1 は 0 である。インター予測モードが L1 予測 (Pred_L1) の場合、L0 予測を利用するか否かを示すフラグ predFlagL0 は 0、L1 予測を利用するか否かを示すフラグ predFlagL1 は 1 である。インター予測モードが双予測 (Pred_BI) の場合、L0 予測を利用するか否かを示すフラグ predFlagL0、L1 予測を利用するか否かを示すフラグ predFlagL1 は共に 1 である。さらに、処理対象の符号化ブロックの予測モード PredMode がインター予測 (PRED_INTER) でマージモードの時に、マージ候補を導出する。符号化情報格納メモリ 205 に記憶されている既に復号された符号化ブロックの符号化情報を用いて、複数のマージの候補を導出して後述するマージ候補リストに登録し、マージ候補リストに登録された複数のマージ候補の中からビット

30

40

50

列復号部 201 で復号され供給されるマージインデックスに対応したマージ候補を選択し、選択されたマージ候補の L0 予測、及び L1 予測を利用するか否かを示すフラグ `predFlagL0[xP][yP]`、`predFlagL1[xP][yP]`、L0、L1 の参照インデックス `refIdxL0[xP][yP]`、`refIdxL1[xP][yP]`、L0、L1 の動きベクトル `mvL0[xP][yP]`、`mvL1[xP][yP]` 等のインター予測情報を符号化情報格納メモリ 205 に格納する。ここで、xP、yP はピクチャ内での符号化ブロックの左上の画素の位置を示すインデックスである。インター予測部の詳細な構成と動作は後述する。

【0041】

イントラ予測部 204 は、処理対象の符号化ブロックの予測モード `PredMode` がイントラ予測 (`PRED_INTRA`) の時に、イントラ予測を行う。ビット列復号部 201 で復号された符号化情報にはイントラ予測モードが含まれており、イントラ予測モードに応じて、復号画像メモリ 208 に格納されている復号済みの画像信号からイントラ予測により予測画像信号を生成し、予測画像信号を復号画像信号重畳部 207 に供給する。イントラ予測部 204 は、画像符号化装置 100 のイントラ予測部 103 に対応するものであるから、イントラ予測部 103 と同様の処理を行う。

【0042】

逆量子化・逆直交変換部 206 は、ビット列復号部 201 で復号された直交変換・量子化された残差信号に対して逆直交変換及び逆量子化を行い、逆直交変換・逆量子化された残差信号を得る。

【0043】

復号画像信号重畳部 207 は、インター予測部 203 でインター予測された予測画像信号、またはイントラ予測部 204 でイントラ予測された予測画像信号と、逆量子化・逆直交変換部 206 により逆直交変換・逆量子化された残差信号とを重畳することにより、復号画像信号を復号し、復号画像メモリ 208 に格納する。復号画像メモリ 208 に格納する際には、復号画像に対して符号化によるブロック歪等を減少させるフィルタリング処理を施した後、復号画像メモリ 208 に格納してもよい。

【0044】

次に、画像符号化装置 100 におけるブロック分割部 101 の動作について説明する。図 3 は、画像をツリーブロックに分割し、各ツリーブロックをさらに分割する動作を示すフローチャートである。まず、入力された画像を、所定サイズのツリーブロックに分割する (ステップ S1001)。各ツリーブロックについては、所定の順序、すなわちラスカスキャン順に走査し (ステップ S1002)、処理対象のツリーブロックの内部を分割する (ステップ S1003)。

【0045】

図 7 は、ステップ S1003 の分割処理の詳細動作を示すフローチャートである。まず、処理対象のブロックを 4 分割するか否かを判断する (ステップ S1101)。

【0046】

処理対象ブロックを 4 分割すると判断した場合は、処理対象ブロックを 4 分割する (ステップ S1102)。処理対象ブロックを分割した各ブロックについて、Z スキャン順、すなわち左上、右上、左下、右下の順に走査する (ステップ S1103)。図 5 は、Z スキャン順の例であり、図 6 の 601 は、処理対象ブロックを 4 分割した例である。図 6 の 601 の番号 0 ~ 3 は処理の順番を示したものである。そしてステップ S1101 で分割した各ブロックについて、図 7 のフローチャートを再帰的に呼び出す。

【0047】

処理対象ブロックを 4 分割しないと判断した場合は、2 - 3 分割を行う (ステップ S1105)。

【0048】

図 8 は、ステップ S1105 の 2 - 3 分割処理の詳細動作を示すフローチャートである。まず、処理対象のブロックを 2 - 3 分割するか否か、すなわち 2 分割または 3 分割の何れかを行うか否かを判断する (ステップ S1201)。

10

20

30

40

50

【 0 0 4 9 】

処理対象ブロックを 2 - 3 分割すると判断しない場合、すなわち分割しないと判断した場合は、分割を終了し（ステップ S 1 2 1 1）、上位階層のブロックに戻る。

【 0 0 5 0 】

処理対象のブロックを 2 - 3 分割すると判断した場合は、さらに処理対象ブロックを 2 分割するか否か（ステップ S 1 2 0 2）を判断する。

【 0 0 5 1 】

処理対象ブロックを 2 分割すると判断した場合は、処理対象ブロックを垂直方向に分割するか否かを判断し（ステップ S 1 2 0 3）、その結果に基づき、処理対象ブロックを垂直方向に分割する（ステップ S 1 2 0 4）か、処理対象ブロックを水平方向に分割する（ステップ S 1 2 0 5）。ステップ S 1 2 0 4 の結果、処理対象ブロックは、図 6 0 2 に示す通り、垂直方向 2 分割に分割され、ステップ S 1 2 0 5 の結果、処理対象ブロックは、図 6 0 4 に示す通り、水平方向 2 分割に分割される。

10

【 0 0 5 2 】

ステップ S 1 2 0 2 において、処理対象のブロックを 2 分割すると判断しなかった場合、すなわち 3 分割すると判断した場合は、処理対象ブロックを垂直方向に分割するか否かを判断し（ステップ S 1 2 0 6）、その結果に基づき、処理対象ブロックを垂直方向に分割する（ステップ S 1 2 0 7）か、処理対象ブロックを水平方向に分割する（ステップ S 1 2 0 8）。ステップ S 1 2 0 7 の結果、処理対象ブロックは、図 6 0 3 に示す通り、垂直方向 3 分割に分割され、ステップ S 1 2 0 8 の結果、処理対象ブロックは、図 6 0 5 に示す通り、水平方向 3 分割に分割される。

20

【 0 0 5 3 】

ステップ S 1 2 0 4 からステップ S 1 2 0 5 のいずれかを実行後、処理対象ブロックを分割した各ブロックについて、左から右、上から下の順に走査する（ステップ S 1 2 0 9）。図 6 の 6 0 2 から 6 0 5 の番号 0 ~ 3 は処理の順番を示したものである。分割した各ブロックについて、図 8 のフローチャートを再帰的に呼び出す。

【 0 0 5 4 】

ここで説明した再帰的なブロック分割は、分割する回数、または、処理対象のブロックのサイズ等により、分割要否を制限してもよい。分割要否を制限する情報は、符号化装置と復号装置の間で予め取り決めを行うことで、情報の伝達を行わない構成で実現してもよい、符号化装置が分割要否を制限する情報を決定し、符号化ビット列に記録することにより、復号装置に伝達する構成で実現してもよい。

30

【 0 0 5 5 】

ここで、あるブロックを分割した場合、分割前のブロックを親ブロックと呼び、分割後の各ブロックを子ブロックと呼ぶ。

【 0 0 5 6 】

次に、画像復号装置 2 0 0 におけるブロック分割部 2 0 2 の動作について説明する。ブロック分割部 2 0 2 は、画像符号化装置 1 0 0 のブロック分割部 1 0 1 と同様の処理手順でツリーブロックを分割するものである。ただし、画像符号化装置 1 0 0 のブロック分割部 1 0 1 では、画像認識による最適形状の推定や歪レート最適化等最適化手法を適用し、最適なブロック分割の形状を決定するのに対し、画像復号装置 2 0 0 におけるブロック分割部 2 0 2 は、符号化ビット列に記録されたブロック分割情報を復号することにより、ブロック分割形状を決定する点が異なる。

40

【 0 0 5 7 】

第 1 の実施の形態のブロック分割に関するシンタックス（符号化ビット列の構文規則）を図 9 に示す。coding_quadtree()はブロックの 4 分割処理にかかるシンタックスを表し、multi_type_tree()はブロックの 2 分割または 3 分割処理にかかるシンタックスを表す。qt_splitはブロックを 4 分割するか否かを示すフラグであり、ブロックを 4 分割する場合は、qt_split=1、4 分割しない場合は、qt_split=0とする。4 分割する場合(qt_split=1)、4 分割した各ブロックについて、再帰的に 4 分割処理をする(coding_quadtree(0),

50

coding_quadtree(1), coding_quadtree(2), coding_quadtree(3))。4分割しない場合($qt_split=0$)は、multi_type_tree()に従い、後続の分割を決定する。mtt_splitは、さらに分割をするか否かを示すフラグである。さらに分割をする場合($mtt_split=1$)、垂直方向に分割するか水平方向に分割するかを示すフラグであるmtt_split_verticalと、2分割するか3分割するかを決定するフラグであるmtt_split_binaryを参照する。mtt_split_vertical=1は、垂直方向に分割することを示し、mtt_split_vertical=0は、水平方向に分割することを示す。mtt_split_binary=1は、2分割することを示し、mtt_split_binary=0は3分割することを示す。mtt_split=0となるまで、再帰的にmulti_type_treeを呼び出すことにより、階層的なブロック分割を行う。

【0058】

< インター予測 >

実施の形態に係るインター予測方法は、図1の動画像符号化装置のインター予測部102および図2の動画像復号装置のインター予測部203において実施される。

【0059】

実施の形態によるインター予測方法について、図面を用いて説明する。インター予測方法は符号化ブロック単位で符号化及び復号の処理の何れでも実施される。

【0060】

< 符号化側のインター予測部102の説明 >

図16は図1の動画像符号化装置のインター予測部102の詳細な構成を示す図である。通常予測動きベクトルモード導出部301は複数の通常予測動きベクトル候補を導出して予測動きベクトルを選択し、検出した動きベクトルとの差分ベクトルを算出する。検出されたインター予測モード、参照インデックス、動きベクトル、算出された差分ベクトルが通常予測動きベクトルモードのインター予測情報となる。このインター予測情報がインター予測モード判定部305に供給される。通常予測動きベクトルモード導出部301の詳細な構成と処理については後述する。

【0061】

通常マージモード導出部302では複数の通常マージ候補を導出して通常マージ候補を選択し、通常マージモードのインター予測情報を得る。このインター予測情報がインター予測モード判定部305に供給される。通常マージモード導出部302の詳細な構成と処理については後述する。

【0062】

サブブロック予測動きベクトルモード導出部303では複数のサブブロック予測動きベクトル候補を導出してサブブロック予測動きベクトルを選択し、検出した動きベクトルとの差分ベクトルを算出する。検出されたインター予測モード、参照インデックス、動きベクトル、算出された差分ベクトルが通常予測動きベクトルモードのインター予測情報となる。このインター予測情報がインター予測モード判定部305に供給される。サブブロック予測動きベクトルモード導出部303の詳細な構成と処理については後述する。

【0063】

サブブロックマージモード導出部304では複数のサブブロックマージ候補を導出してサブブロックマージ候補を選択し、サブブロックマージモードのインター予測情報を得る。このインター予測情報がインター予測モード判定部305に供給される。サブブロックマージモード導出部304の詳細な構成と処理については後述する。

【0064】

インター予測モード判定部305では通常予測動きベクトルモード導出部301、通常マージモード導出部302、サブブロック予測動きベクトルモード導出部303、サブブロックマージモード導出部304から供給されるインター予測情報に基づいて、インター予測情報を判定する。インター予測モード判定部の305から判定結果に応じたインター予測情報が動き補償予測部306に供給される。

【0065】

動き補償予測部306では判定されたインター予測情報に基づいて、復号画像メモリ1

10

20

30

40

50

04に格納されている参照画像信号に対してインター予測を行う。詳細な構成と処理については後述する。

【0066】

<復号側のインター予測部203の説明>

図22は図2の動画像復号装置のインター予測部203の詳細な構成を示す図である。

【0067】

通常予測動きベクトルモード導出部401は複数の通常予測動きベクトル候補を導出して予測動きベクトルを選択し、検出した動きベクトルとの差分ベクトルを算出する。検出されたインター予測モード、参照インデックス、動きベクトル、差分ベクトルが通常予測動きベクトルモードのインター予測情報となる。このインター予測情報がスイッチ408

10

を経由して動き補償予測部406に供給される。通常予測動きベクトルモード導出部401の詳細な構成と処理については後述する。

【0068】

通常マージモード導出部402では複数の通常マージ候補を導出して通常マージ候補を選択し、通常マージモードのインター予測情報を得る。このインター予測情報がスイッチ408を経由して動き補償予測部406に供給される。通常マージモード導出部402の詳細な構成と処理については後述する。

【0069】

サブブロック予測動きベクトルモード導出部403では複数のサブブロック予測動きベクトル候補を導出してサブブロック予測動きベクトルを選択し、検出した動きベクトルとの差分ベクトルを算出する。検出されたインター予測モード、参照インデックス、動きベクトル、算出された差分ベクトルが通常予測動きベクトルモードのインター予測情報となる。このインター予測情報がスイッチ408を経由して動き補償予測部406に供給される。サブブロック予測動きベクトルモード導出部403の詳細な構成と処理については後述する。

20

【0070】

サブブロックマージモード導出部404では複数のサブブロックマージ候補を導出してサブブロックマージ候補を選択し、サブブロックマージモードのインター予測情報を得る。このインター予測情報がスイッチ408を経由して動き補償予測部406に供給される。サブブロックマージモード導出部404の詳細な構成と処理については後述する。

30

【0071】

動き補償予測部406では判定されたインター予測情報に基づいて、復号画像メモリ208に格納されている参照画像信号に対してインター予測を行う。詳細な構成と処理については符号化側と同様である。

【0072】

<通常予測動きベクトルモード導出部(通常AMVP)>

図17の通常予測動きベクトルモード導出部301は、空間予測動きベクトル候補導出部321、時間予測動きベクトル候補導出部322、履歴予測動きベクトル候補導出部323、予測動きベクトル候補補充部325、通常動きベクトル検出部326、予測動きベクトル候補選択部327、動きベクトル減算部328を含む。

40

【0073】

図23の通常予測動きベクトルモード導出部401は、空間予測動きベクトル候補導出部421、時間予測動きベクトル候補導出部422、履歴予測動きベクトル候補導出部423、予測動きベクトル候補補充部425、予測動きベクトル候補選択部426、動きベクトル加算部427を含む。

【0074】

符号化側の通常予測動きベクトルモード導出部301および復号側の通常予測動きベクトルモード導出部401の処理手順について、それぞれ図19、図25のフローチャートを用いて説明する。図19は符号化側の通常動きベクトルモード導出部301による通常予測動きベクトルモード導出処理手順を示すフローチャートであり、図25は復号側の通

50

常動きベクトルモード導出部 401 による通常予測動きベクトルモード導出処理手順を示すフローチャートである。

【0075】

< 通常予測動きベクトルモード導出部（通常AMVP）：符号化側の説明 >

図19を参照して符号化側の通常予測動きベクトルモード導出処理手順を説明する。図19の処理手順の説明において、明細書の動きベクトルという用語と、図19の通常動きベクトルという用語は対応するものとする。

【0076】

まず、通常動きベクトル検出部 326 でインター予測モードおよび参照インデックス毎に通常動きベクトルを検出する（図19のステップ S100）。

10

【0077】

続いて、空間予測動きベクトル候補導出部 321、時間予測動きベクトル候補導出部 322、履歴予測動きベクトル候補導出部 323、予測動きベクトル候補補充部 325、予測動きベクトル候補選択部 327、動きベクトル減算部 328 で、通常予測動きベクトルモードのインター予測で用いる動きベクトルの差分動きベクトルを L0、L1 毎にそれぞれ算出する（図19のステップ S101～S106）。具体的には処理対象ブロックの予測モード PredMode がインター予測（MODE_INTER）で、インター予測モードが L0 予測（Pred_L0）の場合、L0 の予測動きベクトル候補リスト mvpListL0 を算出して、予測動きベクトル mvpL0 を選択し、L0 の動きベクトル mvL0 の差分動きベクトル mvdL0 を算出する。処理対象ブロックのインター予測モードが L1 予測（Pred_L1）の場合、L1 の予測動きベクトル候補リスト mvpListL1 を算出して、予測動きベクトル mvpL1 を選択し、L1 の動きベクトル mvL1 の差分動きベクトル mvdL1 を算出する。処理対象ブロックのインター予測モードが双予測（Pred_BI）の場合、L0 予測と L1 予測が共に行われ、L0 の予測動きベクトル候補リスト mvpListL0 を算出して、L0 の予測動きベクトル mvpL0 を選択し、L0 の動きベクトル mvL0 の差分動きベクトル mvdL0 を算出するとともに、L1 の予測動きベクトル候補リスト mvpListL1 を算出して、L1 の予測動きベクトル mvpL1 を算出し、L1 の動きベクトル mvL1 の差分動きベクトル mvdL1 をそれぞれ算出する。

20

【0078】

L0、L1 それぞれについて、差分動きベクトル算出処理を行うが、L0、L1 とともに共通の処理となる。したがって、以下の説明においては L0、L1 を共通の LX として表す。L0 の差分動きベクトルを算出する処理では X が 0 であり、L1 の差分動きベクトルを算出する処理では X が 1 である。また、LX の差分動きベクトルを算出する処理中に、LX ではなく、もう一方のリストの情報を参照する場合、もう一方のリストを LY として表す。

30

【0079】

LX の動きベクトル mvLX を使用する場合（図19のステップ S102：YES）、LX の予測動きベクトルの候補を算出して LX の予測動きベクトル候補リスト mvpListLX を構築する（図19のステップ S103）。通常予測動きベクトルモード導出部 301 の中の空間予測動きベクトル候補導出部 321、時間予測動きベクトル候補導出部 322、履歴予測動きベクトル候補導出部 323、予測動きベクトル候補補充部 325 で複数の予測動きベクトルの候補を導出して予測動きベクトル候補リスト mvpListLX を構築する。図19のステップ S103 の詳細な処理手順については図20のフローチャートを用いて後述する。

40

【0080】

続いて、予測動きベクトル候補選択部 327 により、LX の予測動きベクトル候補リスト mvpListLX から LX の予測動きベクトル mvLX を選択する（図19のステップ S104）。動きベクトル mvLX と予測動きベクトル候補リスト mvpListLX の中に格納された各予測動きベクトルの候補 mvpListLX[i] との差分であるそれぞれの差分動きベクトルを算出する。それら差分動きベクトルを符号化したときの符号量を予測動きベクトル候補リスト mvpListLX の要素（予測動きベクトル候補）ごとに算出する。そして、予測動きベクトル

50

ル候補リストmvpListLXに登録された各要素の中で、予測動きベクトルの候補毎の符号量が最小となる予測動きベクトルの候補mvpListLX[i]を予測動きベクトルmvpLXとして選択し、そのインデックスiを取得する。予測動きベクトル候補リストmvpListLXの中で最小の発生符号量となる予測動きベクトルの候補が複数存在する場合には、予測動きベクトル候補リストmvpListLXの中のインデックスiが小さい番号で表される予測動きベクトルの候補mvpListLX[i]を最適予測動きベクトルmvpLXとして選択し、そのインデックスiを取得する。

【 0 0 8 1 】

続いて、動きベクトル減算部 3 2 8 で、L X の動きベクトルmvLXから選択された L X の予測動きベクトルmvpLXを減算し、

$$mvdLX = mvLX - mvpLX$$

として L X の差分動きベクトルmvdLXを算出する（図 1 9 のステップ S 1 0 5 ）。

【 0 0 8 2 】

< 通常予測動きベクトルモード導出部（通常AMVP）：復号側の説明 >

次に、図 2 5 を参照して復号側の通常予測動きベクトルモード処理手順を説明する。復号側では、空間予測動きベクトル候補導出部 4 2 1、時間予測動きベクトル候補導出部 4 2 2、履歴予測動きベクトル候補導出部 4 2 3、予測動きベクトル候補補充部 4 2 5 で、通常予測動きベクトルモードのインター予測で用いる動きベクトルを L 0、L 1 毎にそれぞれ算出する（図 2 5 のステップ S 2 0 1 ~ S 2 0 6）。具体的には処理対象ブロックの予測モードPredModeがインター予測（MODE_INTER）で、処理対象ブロックのインター予測モードが L 0 予測（Pred_L0）の場合、L 0 の予測動きベクトル候補リストmvpListL0を算出して、予測動きベクトルmvpL0を選択し、L 0 の動きベクトルmvL0を算出する。処理対象ブロックのインター予測モードが L 1 予測（Pred_L1）の場合、L 1 の予測動きベクトル候補リストmvpListL1を算出して、予測動きベクトルmvpL1を選択し、L 1 の動きベクトルmvL1を算出する。処理対象ブロックのインター予測モードが双予測（Pred_BI）の場合、L 0 予測と L 1 予測が共に行われ、L 0 の予測動きベクトル候補リストmvpListL0を算出して、L 0 の予測動きベクトルmvpL0を選択し、L 0 の動きベクトルmvL0を算出するとともに、L 1 の予測動きベクトル候補リストmvpListL1を算出して、L 1 の予測動きベクトルmvpL1を算出し、L 1 の動きベクトルmvL1をそれぞれ算出する。

【 0 0 8 3 】

符号化側と同様に、復号側でも L 0、L 1 それぞれについて、動きベクトル算出処理を行うが、L 0、L 1 とともに共通の処理となる。したがって、以下の説明においては L 0、L 1 を共通の L X として表す。L X は処理対象の符号化ブロックのインター予測に用いるインター予測モードを表す。L 0 の動きベクトルを算出する処理では X が 0 であり、L 1 の動きベクトルを算出する処理では X が 1 である。また、L X の動きベクトルを算出する処理中に、算出対象の L X と同じ参照リストではなく、もう一方の参照リストの情報を参照する場合、もう一方の参照リストを L Y として表す。

【 0 0 8 4 】

L X の動きベクトルmvLXを使用する場合（図 2 5 のステップ S 2 0 2：YES）、L X の予測動きベクトルの候補を算出して L X の予測動きベクトル候補リストmvpListLXを構築する（図 2 5 のステップ S 2 0 3）。通常予測動きベクトルモード導出部 4 0 1 の中の空間予測動きベクトル候補導出部 4 2 1、時間予測動きベクトル候補導出部 4 2 2、履歴予測動きベクトル候補導出部 4 2 3、予測動きベクトル候補補充部 4 2 5 で複数の予測動きベクトルの候補を算出し、予測動きベクトル候補リストmvpListLXを構築する。図 2 5 のステップ S 2 0 3 の詳細な処理手順については図 2 0 のフローチャートを用いて後述する。

【 0 0 8 5 】

続いて、予測動きベクトル候補選択部 4 2 6 で予測動きベクトル候補リストmvpListLXからビット列復号部 2 0 1 にて復号されて供給される予測動きベクトルのインデックスmv

10

20

30

40

50

pIdxLXに対応する予測動きベクトルの候補mvplListLX[mvpIdxLX]を選択された予測動きベクトルmvplXとして取り出す（図25のステップS204）。

【0086】

続いて、動きベクトル加算部427でビット列復号部201にて復号されて供給されるLXの差分動きベクトルmvdLXとLXの予測動きベクトルmvplXを加算し、

$$mvLX = mvplX + mvdLX$$

としてLXの動きベクトルmvLXを算出する（図25のステップS205）。

【0087】

＜通常予測動きベクトルモード導出部（通常AMVP）：動きベクトルの予測方法＞

図20は本発明の実施の形態に係る動画像符号化装置の通常予測動きベクトルモード導出部301及び動画像復号装置の通常予測動きベクトルモード導出部401とで共通する機能を有する通常予測動きベクトルモード導出処理の処理手順を表すフローチャートである。

【0088】

通常予測動きベクトルモード導出部301及び通常予測動きベクトルモード導出部401では、予測動きベクトル候補リストmvplListLXを備えている。予測動きベクトル候補リストmvplListLXはリスト構造を成し、予測動きベクトル候補リスト内部の所在を示す予測動きベクトルインデックスと、インデックスに対応する予測動きベクトル候補を要素として格納する記憶領域が設けられている。予測動きベクトルインデックスの数字は0から開始され、予測動きベクトル候補リストmvplListLXの記憶領域に、予測動きベクトル候補が格納される。本実施の形態においては、予測動きベクトル候補リストmvplListLXは少なくとも2個の予測動きベクトル候補（インター予測情報）を登録することができるものとする。さらに、予測動きベクトル候補リストmvplListLXに登録されている予測動きベクトル候補数を示す変数numCurrMvpCandに0を設定する。

【0089】

空間予測動きベクトル候補導出部321及び421は、左側に隣接するブロックからの予測動きベクトルの候補を導出する。この処理では、左側に隣接するブロック（A0またはA1）の予測動きベクトル候補が利用できるか否かを示すフラグavailableFlagLXA、及び動きベクトルmvLXA、参照インデックスrefIdxAを導出し、mvLXAを予測動きベクトル候補リストmvplListLXに追加する（図20のステップS301）。なお、L0のときXは0、L1のときXは1とする（以下同様）。続いて、空間予測動きベクトル候補導出部321及び421は、上側に隣接するブロック（B0、B1またはB2）からの予測動きベクトルの候補を導出する。この処理では、上側に隣接するブロックの予測動きベクトル候補が利用できるか否かを示すフラグavailableFlagLXB、及び動きベクトルmvLXB、参照インデックスrefIdxBを導出し、mvLXAとmvLXBが等しくなければ、mvLXBを予測動きベクトル候補リストmvplListLXに追加する（図20のステップS302）。図20のステップS301とS302の処理は参照する隣接ブロックの位置と数が異なる点以外は共通であり、符号化ブロックの予測動きベクトル候補が利用できるか否かを示すフラグavailableFlagLXN、及び動きベクトルmvLXN、参照インデックスrefIdxN（NはAまたはB、以下同様）を導出する。

【0090】

続いて、時間予測動きベクトル候補導出部322及び422は、現在の処理対象ピクチャとは時間が異なるピクチャにおける符号化ブロックからの予測動きベクトルの候補を導出する。この処理では、異なる時間のピクチャにおける符号化ブロックの予測動きベクトル候補が利用できるか否かを示すフラグavailableFlagLXCol、及び動きベクトルmvLXCol、参照インデックスrefIdxCol、参照リストlistColを導出し、mvLXColを予測動きベクトル候補リストmvplListLXに追加する（図20のステップS303）。このステップS303の導出処理手順を後ほど詳細に説明する。

【0091】

なお、シーケンス（SPS）、ピクチャ（PPS）またはスライスの単位で時間予測動きベク

10

20

30

40

50

トル候補導出部 3 2 2 及び 4 2 2 の処理を省略することができるものとする。

【 0 0 9 2 】

続いて、履歴予測動きベクトル候補導出部 3 2 3 及び 4 2 3 は履歴予測動きベクトル候補リストHmvpCandListに登録されている履歴予測動きベクトル候補を予測動きベクトル候補リストmvpListLXに追加する。(図 2 0 のステップ S 3 0 4)。このステップ S 3 0 4 の登録処理手順については図 4 1 のフローチャートを用いて後ほど詳細に説明する。

【 0 0 9 3 】

続いて予測動きベクトル候補補充部 3 2 5 及び 4 2 5 は予測動きベクトル候補リストmvpListLXを満たすまで (0 , 0) 等、所定の値の動きベクトルを追加する (図 2 0 のステップ S 3 0 5) 。

【 0 0 9 4 】

< 通常マージモード導出部 (通常マージ) >

図 1 8 の通常マージモード導出部 3 0 2 は、空間マージ候補導出部 3 4 1、時間マージ候補導出部 3 4 2、平均マージ候補導出部 3 4 4、履歴マージ候補導出部 3 4 5、マージ候補補充部 3 4 6、マージ候補選択部 3 4 7 を含む。

【 0 0 9 5 】

図 2 4 の通常マージモード導出部 4 0 2 は、空間マージ候補導出部 4 4 1、時間マージ候補導出部 4 4 2、平均マージ候補導出部 4 4 4、履歴マージ候補導出部 4 4 5、マージ候補補充部 4 4 6、マージ候補選択部 4 4 7 を含む。

【 0 0 9 6 】

図 2 1 は本発明の実施の形態に係る動画像符号化装置の通常マージモード導出部 3 0 2 及び動画像復号装置の通常マージモード導出部 4 0 2 とで共通する機能を有する通常マージモード導出処理の手順を説明するフローチャートである。

【 0 0 9 7 】

以下、諸過程を順を追って説明する。なお、以下の説明においては特に断りのない限りスライスタイプslice_typeがBスライスの場合について説明するが、Pスライスの場合にも適用できる。ただし、スライスタイプslice_typeがPスライスの場合、インター予測モードとしてL 0 予測 (Pred_L0) だけがあり、L 1 予測 (Pred_L1)、双予測 (Pred_BI) がないので、L 1 に纏わる処理を省略することができる。

【 0 0 9 8 】

通常マージモード導出部 3 0 2 及び通常マージモード導出部 4 0 2 では、マージ候補リストmergeCandListを備えている。マージ候補リストmergeCandListはリスト構造を成し、マージ候補リスト内部の所在を示すマージインデックスと、インデックスに対応するマージ候補を要素として格納する記憶領域が設けられている。マージインデックスの数字は 0 から開始され、マージ候補リストmergeCandListの記憶領域に、マージ候補が格納される。以降の処理では、マージ候補リストmergeCandListに登録されたマージインデックス i のマージ候補は、mergeCandList[i]で表すこととする。本実施の形態においては、マージ候補リストmergeCandListは少なくとも 6 個のマージ候補 (インター予測情報) を登録することができるものとする。さらに、マージ候補リストmergeCandListに登録されているマージ候補数を示す変数numCurrMergeCandに 0 を設定する。

【 0 0 9 9 】

空間マージ候補導出部 3 4 1 及び空間マージ候補導出部 4 4 1 では、動画像符号化装置の符号化情報格納メモリ 1 1 1 または動画像復号装置の符号化情報格納メモリ 2 0 5 に格納されている符号化情報から、処理対象ブロックの左側と上側に隣接するブロックからの空間マージ候補 A , B を導出して、導出された空間マージ候補をマージ候補リストmergeCandListに登録する (図 2 1 のステップ S 4 0 1)。ここで、空間マージ候補 A , B または時間マージ候補 C o 1 のいずれかを示す N を定義する。ブロック N のインター予測情報が空間マージ候補 N として利用できるか否かを示すフラグavailableFlagN、空間マージ候補 N の L 0 の参照インデックスrefIdxL0N及び L 1 の参照インデックスrefIdxL1N、L 0 予測が行われるか否かを示す L 0 予測フラグpredFlagL0Nおよび L 1 予測が行われるか否

10

20

30

40

50

かを示す L 1 予測フラグpredFlagL1N、L 0 の動きベクトルmvL0N、L 1 の動きベクトルmvL1Nを導出する。ただし、本実施の形態においては処理対象となる符号化ブロックを含むブロックに含まれる他の符号化ブロックを参照せずに、マージ候補を導出するので、処理対象の符号化ブロックを含むブロックに含まれる空間マージ候補は導出しない。

【0100】

続いて、時間マージ候補導出部342及び時間マージ候補導出部442では、異なる時間のピクチャからの時間マージ候補を導出して、導出された時間マージ候補をマージ候補リストmergeCandListに登録する(図21のステップS402)。時間マージ候補が利用できるか否かを示すフラグavailableFlagCol、時間マージ候補のL0予測が行われるか否かを示すL0予測フラグpredFlagL0ColおよびL1予測が行われるか否かを示すL1予測フラグpredFlagL1Col、及びL0の動きベクトルmvL0Col、L1の動きベクトルmvL1Colを導出する。ステップS402の詳細な処理手順については後ほど図56を参照して詳細に説明する。

10

【0101】

なお、シーケンス(SPS)、ピクチャ(PPS)またはスライスの単位で時間マージ候補導出部342及び時間マージ候補導出部442の処理を省略することができるものとする。

【0102】

続いて、履歴マージ候補導出部345及び履歴マージ候補導出部445では、履歴予測動きベクトル候補リストHmvpCandListに登録されている履歴予測動きベクトル候補をマージ候補リストmergeCandListに追加する(図21のステップS403)。ステップS403の詳細な処理手順については図62のフローチャートを用いて後ほど詳細に説明する。

20

【0103】

続いて、平均マージ候補導出部344及び平均マージ候補導出部444では、マージ候補リストmergeCandListから平均マージ候補を導出して、導出された平均マージ候補をマージ候補リストmergeCandListに登録する(図21のステップS404)。ステップS404の詳細な処理手順については図41のフローチャートを用いて後ほど詳細に説明する。

【0104】

続いて、マージ候補補充部346及びマージ候補補充部446では、マージ候補リストmergeCandList内に登録されているマージ候補数numCurrMergeCandが、最大マージ候補数MaxNumMergeCandより小さい場合、マージ候補リストmergeCandList内に登録されているマージ候補数numCurrMergeCandが最大マージ候補数MaxNumMergeCandを上限として追加マージ候補を導出して、マージ候補リストmergeCandListに登録する(図21のステップS405)。最大マージ候補数MaxNumMergeCandを上限として、Pスライスでは、異なる参照インデックスで動きベクトルが(0, 0)の値を持つ予測モードがL0予測(Pred_L0)のマージ候補を追加する。Bスライスでは、異なる参照インデックスで動きベクトルが(0, 0)の値を持つ予測モードが双予測(Pred_BI)のマージ候補を追加する。

30

【0105】

続いて、マージ候補選択部347及びマージ候補選択部447では、マージ候補リストmergeCandList内に登録されているマージ候補からマージ候補を選択する。符号化側のマージ候補選択部347では、符号量とひずみ量を算出することによりマージ候補を選択し、選択されたマージ候補を示すマージインデックス、マージ候補のインター予測情報を動き補償予測部306に供給する。一方、復号側のマージ候補選択部447では、復号されたマージインデックスに基づいて、マージ候補を選択し、選択されたマージ候補を動き補償予測部406に供給する。

40

【0106】

通常マージモード導出部302及び通常マージモード導出部402は、ある符号化ブロックのサイズ(幅と高さの積)が32未満の場合、その符号化ブロックの親ブロックにおいてマージ候補が導出される。そして、全ての子ブロックでは、親ブロックにおいて導出されたマージ候補を用いる。ただし、親ブロックのサイズが32以上で、かつ画面内に収

50

まっている場合に限る。

【0107】

<サブブロック予測動きベクトルモード導出>

サブブロック予測動きベクトルモード導出について説明する。

【0108】

図26は、本実施の形態の符号化装置におけるサブブロック予測動きベクトルモード導出部303のブロック図である。

【0109】

まず、アフィン継承予測動きベクトル候補導出部361において、アフィン継承予測動きベクトル候補を導出する。アフィン継承予測動きベクトル候補導出の詳細については後述する。

10

【0110】

続いて、アフィン構築予測動きベクトル候補導出部362において、アフィン構築予測動きベクトル候補を導出する。アフィン構築予測動きベクトル候補導出の詳細については後述する。

【0111】

続いて、アフィン同一予測動きベクトル候補導出部363において、アフィン同一予測動きベクトル候補を導出する。アフィン同一予測動きベクトル候補導出の詳細については後述する。

【0112】

サブブロック動きベクトル検出部366は、サブブロック予測動きベクトルモードに適するサブブロック動きベクトルを検出し、検出したベクトルをサブブロック予測動きベクトル候補選択部367、差分演算部368に供給する。

20

【0113】

サブブロック予測動きベクトル候補選択部367は、アフィン継承予測動きベクトル候補導出部361、アフィン構築予測動きベクトル候補導出部362、アフィン同一予測動きベクトル候補導出部363において導出されたサブブロック予測動きベクトル候補の中から、サブブロック動きベクトル検出部366から供給された動きベクトルに基づいて、サブブロック予測動きベクトル候補を選択し、選択されたサブブロック予測動きベクトル候補に関する情報をインター予測モード判定部305、差分演算部368に供給する。

30

【0114】

差分演算部368は、サブブロック動きベクトル検出部366から供給された動きベクトルベクトルから、サブブロック予測動きベクトル候補選択部367で選択されたサブブロック予測動きベクトルを減算した差分予測動きベクトルを、インター予測モード判定部305に供給する。

【0115】

図27は、本実施の形態の復号装置におけるサブブロック予測動きベクトルモード導出部403のブロック図である。

【0116】

まず、アフィン継承予測動きベクトル候補導出部461において、アフィン継承予測動きベクトル候補を導出する。アフィン継承予測動きベクトル候補導出部461の処理は本実施の形態の符号化装置におけるアフィン継承予測動きベクトル候補導出部361の処理と同一である。

40

【0117】

続いて、アフィン構築予測動きベクトル候補導出部462において、アフィン構築予測動きベクトル候補を導出する。アフィン構築予測動きベクトル候補導出部462の処理は本実施の形態の符号化装置におけるアフィン構築予測動きベクトル候補導出部362の処理と同一である。

【0118】

続いて、アフィン同一予測動きベクトル候補導出部463において、アフィン同一予測

50

動きベクトル候補を導出する。アフィン同一予測動きベクトル候補導出部 4 6 3 の処理は本実施の形態の符号化装置におけるアフィン同一予測動きベクトル候補導出部 3 6 3 の処理と同一である。

【 0 1 1 9 】

サブブロック予測動きベクトル候補選択部 4 6 6 は、アフィン継承予測動きベクトル候補導出部 4 6 1、アフィン構築予測動きベクトル候補導出部 4 6 2、アフィン同一予測動きベクトル候補導出部 4 6 3 において導出されたサブブロック予測動きベクトル候補の中から、符号化装置から伝送され復号される予測動きベクトルインデックスに基づいて、サブブロック予測動きベクトル候補を選択し、選択されたサブブロック予測動きベクトル候補に関する情報を動き補償予測部 4 0 6、加算演算部 4 6 7 に供給する。

10

【 0 1 2 0 】

加算演算部 4 6 7 は、サブブロック予測動きベクトル候補選択部 4 6 6 で選択されたサブブロック予測動きベクトルに、符号化装置から伝送され復号される差分動きベクトルを加算して生成した動きベクトルを動き補償予測部 4 0 6 に供給する。

【 0 1 2 1 】

< アフィン継承予測動きベクトル候補導出 >

アフィン継承予測動きベクトル候補導出部 3 6 1 について説明する。アフィン継承予測動きベクトル候補導出部 4 6 1 についてもアフィン継承予測動きベクトル候補導出部 3 6 1 と同様である。

【 0 1 2 2 】

アフィン継承予測動きベクトル候補は、制御点の動きベクトル情報を継承する。

20

【 0 1 2 3 】

図 3 0 は、アフィン継承予測動きベクトル候補導出を説明する図である。

【 0 1 2 4 】

アフィン継承予測動きベクトル候補は、空間的に隣接する符号化済・復号済ブロックの有する制御点の動きベクトルを探索することで得られる。

【 0 1 2 5 】

具体的には、処理対象のブロックの左側に隣接するブロック (A 0 , A 1) と、処理対象のブロックの上側に隣接するブロック (B 0 , B 1 , B 2) から、それぞれ最大 1 つのアフィンモードを探索し、アフィン継承予測動きベクトルとする。

30

【 0 1 2 6 】

図 3 4 は、アフィン継承予測動きベクトル候補導出のフローチャートである。

【 0 1 2 7 】

まず、処理対象のブロックの左側に隣接するブロック (A 0 , A 1) を左グループとし (ステップ S 3 1 0 1)、A 0 を含むブロックがアフィン変換動き補償を用いたブロック (アフィンモード) であるか否かを判断する (ステップ S 3 1 0 2)。A 0 がアフィンモードである場合 (ステップ S 3 1 0 2 : Y E S)、A 0 が使用したアフィンモードを取得し (ステップ S 3 1 0 3)、上側に隣接するブロックの処理に移る。A 0 がアフィンモードでない場合 (ステップ S 3 1 0 2 : N O)、アフィン継承予測動きベクトル候補導出の対象を A 0 - > A 1 とし、A 1 を含むブロックからアフィンモードの取得を試みる。

40

【 0 1 2 8 】

続いて、処理対象のブロックの上側に隣接するブロック (B 0 , B 1 , B 2) を上グループとし (ステップ S 3 1 0 4)、B 0 を含むブロックがアフィンモードであるか否かを判断する (ステップ S 3 1 0 5)。B 0 がアフィンモードである場合 (ステップ S 3 1 0 5 : Y E S)、B 0 が使用したアフィンモードを取得し (ステップ S 3 1 0 6)、処理を終了する。B 0 がアフィンモードでない場合 (ステップ S 3 1 0 5 : N O)、アフィン継承予測動きベクトル候補導出の対象を B 0 - > B 1 とし、B 1 を含むブロックからアフィンモードの取得を試みる。さらに、B 1 がアフィンモードでない場合 (ステップ S 3 1 0 5 : N O)、アフィン継承予測動きベクトル候補導出の対象を B 1 - > B 2 とし、B 2 を含むブロックからアフィンモードの取得を試みる。

50

【 0 1 2 9 】

このように、左側ブロックと上側ブロックにグループを分けて、左側ブロックについては、左下から左上のブロックの順にアフィンモードを探索し、左側ブロックについては、右上から左上のブロックの順にアフィンモードを探索することで、可能な限り異なる2つのアフィンモードを取得することができ、アフィン予測動きベクトルのいずれかが差分動きベクトルの小さくなるアフィン予測動きベクトル候補を導出することができる。

【 0 1 3 0 】

<アフィン構築予測動きベクトル候補導出>

アフィン構築予測動きベクトル候補導出部362について説明する。アフィン構築予測動きベクトル候補導出部462についてもアフィン構築予測動きベクトル候補導出部362と同様である。

10

【 0 1 3 1 】

アフィン構築予測動きベクトル候補は、空間的に隣接するブロックの動き情報から制御点の動きベクトル情報を構築する。

【 0 1 3 2 】

図31は、アフィン構築予測動きベクトル候補導出を説明する図である。

【 0 1 3 3 】

アフィン構築予測動きベクトル候補は、空間的に隣接する符号化済・復号済ブロックの有する動きベクトルを組み合わせることで新たなアフィンモードを構築することで得られる。

【 0 1 3 4 】

20

具体的には、処理対象のブロックの左上側に隣接するブロック(B2, B3, A2)から左上制御点CP0の動きベクトルを導出し、処理対象のブロックの右上側に隣接するブロック(B1, B0)から右上制御点CP1の動きベクトルを導出し、処理対象のブロックの右下側に隣接するブロック(A1, A0)から右下制御点CP2の動きベクトルを導出する。

【 0 1 3 5 】

図35は、アフィン構築予測動きベクトル候補導出のフローチャートである。

【 0 1 3 6 】

まず、左上制御点CP0、右上制御点CP1、右下制御点CP2を導出する(ステップS3201)。左上制御点CP0は、処理対象のブロックと同一の参照画像をもつ参照ブロックを、B2、B3、A2参照ブロックの優先順で探索することで算出される。右上制御点CP1は、処理対象のブロックと同一の参照画像をもつ参照ブロックを、B1、B0参照ブロックの優先順で探索することで算出される。右下制御点CP2は、処理対象のブロックと同一の参照画像をもつ参照ブロックを、A1、A0参照ブロックの優先順で探索することで算出される。

30

【 0 1 3 7 】

アフィン構築予測動きベクトルとして、制御点3本モードを選択する場合(ステップS3202: YES)、3つの制御点(CP0, CP1, CP2)がすべて導出されたか否かを判断する(ステップS3203)。3つの制御点(CP0, CP1, CP2)がすべて導出された場合(ステップS3203: YES)、3つの制御点(CP0, CP1, CP2)を用いたアフィンモデルをアフィン構築予測動きベクトルとする(ステップS3204)。制御点3本モードを選択せず、制御点2本モードを選択した場合(ステップS3202: NO)、2つの制御点(CP0, CP1)がすべて導出されたか否かを判断する(ステップS3205)。2つの制御点(CP0, CP1)がすべて導出された場合(ステップS3205: YES)、2つの制御点(CP0, CP1)を用いたアフィンモデルをアフィン構築予測動きベクトルとする(ステップS3206)。

40

【 0 1 3 8 】

<アフィン同一予測動きベクトル候補導出>

アフィン同一予測動きベクトル候補導出部363について説明する。アフィン同一予測動きベクトル候補導出部463についてもアフィン同一予測動きベクトル候補導出部363

50

3と同様である。

【0139】

アフィン同一予測動きベクトル候補は、各制御点で同一の動きベクトルを導出することで得られる。

【0140】

具体的には、アフィン構築予測動きベクトル候補導出部362・462と同様に、各制御点情報を導出し、すべての制御点をCP0～CP2のいずれかで同一に設定することで得られる。また、通常予測動きベクトルモードと同様に導出した時間動きベクトルをすべての制御点に設定することでも得られる。

【0141】

＜サブブロックマージモード導出＞

サブブロックマージモード導出について説明する。

【0142】

図28は、本実施の形態の符号化装置におけるサブブロックマージモード導出部304のブロック図である。サブブロックマージモード導出部304は、サブブロックマージ候補リストsubblockMergeCandListを備えている。これは、通常マージモード導出部302におけるマージ候補リストmergeCandListと同様にリスト構造を成し、サブブロックマージ候補リスト内部の所在を示すマージインデックスと、インデックスに対応するサブブロックマージ候補を要素として格納する記憶領域が設けられている。本実施の形態においては、サブブロックマージ候補リストsubblockMergeCandListは少なくとも5個のマージ候補（インター予測情報）を登録することができるものとする。ただし、各マージ候補は、さらにサブブロック単位の動きベクトル情報を持つか、あるいは制御点の動きベクトル情報を持つ。

【0143】

まず、サブブロック時間マージ候補導出部381において、サブブロック時間マージ候補を導出する。サブブロック時間マージ候補導出の詳細については後述する。

【0144】

続いて、アフィン継承マージ候補導出部382において、アフィン継承マージ候補を導出する。アフィン継承マージ候補導出の詳細については後述する。

【0145】

続いて、アフィン構築マージ候補導出部383において、アフィン構築マージ候補を導出する。アフィン構築マージ候補導出の詳細については後述する。

【0146】

続いて、アフィン固定マージ候補導出部385において、アフィン固定マージ候補を導出する。アフィン固定マージ候補導出の詳細については後述する。

【0147】

サブブロックマージ候補選択部386は、サブブロック時間マージ候補導出部381、アフィン継承マージ候補導出部382、アフィン構築マージ候補導出部383、アフィン固定マージ候補導出部385において導出されたサブブロックマージ候補の中から、サブブロックマージ候補を選択し、選択されたサブブロックマージ候補に関する情報をインター予測モード判定部305に供給する。

【0148】

図29は、本実施の形態の復号装置におけるサブブロックマージモード導出部404のブロック図である。サブブロックマージモード導出部404は、サブブロックマージ候補リストsubblockMergeCandListを備えている。これは、サブブロックマージモード導出部304と同じものである。

【0149】

まず、サブブロック時間マージ候補導出部481において、サブブロック時間マージ候補を導出する。サブブロック時間マージ候補導出部481の処理はサブブロック時間マージ候補導出部381の処理と同一である。

10

20

30

40

50

【 0 1 5 0 】

続いて、アフィン継承マージ候補導出部 4 8 2 において、アフィン継承マージ候補を導出する。アフィン継承マージ候補導出部 4 8 2 の処理はアフィン継承マージ候補導出部 3 8 2 の処理と同一である。

【 0 1 5 1 】

続いて、アフィン構築マージ候補導出部 4 8 3 において、アフィン構築マージ候補を導出する。アフィン構築マージ候補導出部 4 8 3 の処理はアフィン構築マージ候補導出部 3 8 3 の処理と同一である。

【 0 1 5 2 】

続いて、アフィン固定マージ候補導出部 4 8 5 において、アフィン固定マージ候補を導出する。アフィン固定マージ候補導出部 4 8 5 の処理はアフィン固定マージ候補導出部 4 8 5 の処理と同一である。

【 0 1 5 3 】

サブブロックマージ候補選択部 4 8 6 は、サブブロック時間マージ候補導出部 4 8 1、アフィン継承マージ候補導出部 4 8 2、アフィン構築マージ候補導出部 4 8 3、アフィン固定マージ候補導出部 4 8 5 において導出されたサブブロックマージ候補の中から、符号化装置から伝送され復号されるインデックスに基づいて、サブブロックマージ候補を選択し、選択されたサブブロックマージ候補に関する情報を動き補償予測部 4 0 6 に供給する。

【 0 1 5 4 】

サブブロックマージモード導出部 3 0 4 及びサブブロックマージモード導出部 4 0 4 は、ある符号化ブロックのサイズ（幅と高さの積）が 3 2 未満の場合、その符号化ブロックの親ブロックにおいてサブブロックマージ候補が導出される。そして、全ての子ブロックでは、親ブロックにおいて導出されたサブブロックマージ候補を用いる。ただし、親ブロックのサイズが 3 2 以上で、かつ画面内に収まっている場合に限る。

【 0 1 5 5 】

< サブブロック時間マージ候補導出 >

サブブロック時間マージ候補導出部 3 8 1 の動作については後述する。

【 0 1 5 6 】

< アフィン継承マージ候補導出 >

アフィン継承マージ候補導出部 3 8 2 について説明する。アフィン継承マージ候補導出部 4 8 2 についてもアフィン継承マージ候補導出部 3 8 2 と同様である。

【 0 1 5 7 】

アフィン継承マージ候補は、空間的に隣接するブロックの有するアフィンモデルから制御点のアフィンモデルを継承する。

【 0 1 5 8 】

図 3 2 は、アフィン継承マージ候補導出を説明する図である。アフィンマージ継承マージモード候補の導出は、アフィン継承予測動きベクトルの導出と同様に、空間的に隣接する符号化済・復号済ブロックの有する制御点の動きベクトルを探索することで得られる。

【 0 1 5 9 】

具体的には、処理対象のブロックの左側に隣接するブロック（A 0 , A 1 ）と、処理対象のブロックの上側に隣接するブロック（B 0 , B 1 , B 2 ）から、それぞれ最大 1 つのアフィンモードを探索し、アフィンマージモードに使用する。

【 0 1 6 0 】

図 3 6 は、アフィン継承マージ候補導出のフローチャートである。

【 0 1 6 1 】

まず、処理対象のブロックの左側に隣接するブロック（A 0 , A 1 ）を左グループとし（ステップ S 3 3 0 1 ）、A 0 を含むブロックがアフィンモードであるか否かを判断する（ステップ S 3 3 0 2 ）。A 0 がアフィンモードである場合（ステップ S 3 1 0 2 : Y E S ）、A 0 が使用したアフィンモデルを取得し（ステップ S 3 3 0 3 ）、上側に隣接するブロックの処理に移る。A 0 がアフィンモードでない場合（ステップ S 3 3 0 2 : N O ）

10

20

30

40

50

、アフィン継承マージ候補導出の対象を $A_0 \rightarrow A_1$ とし、 A_1 を含むブロックからアフィンモードの取得を試みる。

【0162】

続いて、処理対象のブロックの上側に隣接するブロック (B_0, B_1, B_2) を上グループとし (ステップ S3304)、 B_0 を含むブロックがアフィンモードであるか否かを判断する (ステップ S3305)。 B_0 がアフィンモードである場合 (ステップ S3305: YES)、 B_0 が使用したアフィンモデルを取得し (ステップ S3306)、処理を終了する。 B_0 がアフィンモードでない場合 (ステップ S3305: NO)、アフィン継承マージ候補導出の対象を $B_0 \rightarrow B_1$ とし、 B_1 を含むブロックからアフィンモードの取得を試みる。さらに、 B_1 がアフィンモードでない場合 (ステップ S3305: NO)、アフィン継承マージ候補導出の対象を $B_1 \rightarrow B_2$ とし、 B_2 を含むブロックからアフィンモードの取得を試みる。

10

【0163】

＜アフィン構築マージ候補導出＞

アフィン構築マージ候補導出部 383 について説明する。アフィン構築マージ候補導出部 483 についてもアフィン構築マージ候補導出部 383 と同様である。

【0164】

図 33 は、アフィン構築マージ候補導出を説明する図である。アフィン構築マージ候補は、空間的に隣接するブロックの有する動き情報及び時間符号化ブロックから制御点のアフィンモデルを構築する。

20

【0165】

具体的には、処理対象のブロックの左上側に隣接するブロック (B_2, B_3, A_2) から左上制御点 CP_0 の動きベクトルを導出し、処理対象のブロックの右上側に隣接するブロック (B_1, B_0) から右上制御点 CP_1 の動きベクトルを導出し、処理対象のブロックの左下側に隣接するブロック (A_1, A_0) から左下制御点 CP_2 の動きベクトルを導出し、処理対象のブロックとの右下側に隣接する符号化ブロック (T_0) から右下制御点 CP_3 の動きベクトルを導出する。

【0166】

図 37 は、アフィン構築マージ候補導出のフローチャートである。

【0167】

まず、左上制御点 CP_0 、右上制御点 CP_1 、左下制御点 CP_2 、右下制御点 CP_3 を導出する (ステップ S3401)。左上制御点 CP_0 は、動き情報を有するブロックを、 B_2, B_3, A_2 ブロックの優先順で探索することで算出される。右上制御点 CP_1 は、動き情報を有するブロックを、 B_1, B_0 ブロックの優先順で探索することで算出される。左下制御点 CP_2 は、動き情報を有するブロックを、 A_1, A_0 ブロックの優先順で探索することで算出される。右下制御点 CP_3 は、時間ブロックの動き情報を探索することで算出される。

30

【0168】

続いて、導出された CP_0, CP_1, CP_2 により 3 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S3402)、構築可能である場合 (ステップ S3402: YES)、 CP_0, CP_1, CP_2 による 3 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S3403)。

40

【0169】

続いて、導出された CP_0, CP_1, CP_3 により 3 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S3404)、構築可能である場合 (ステップ S3404: YES)、 CP_0, CP_1, CP_3 による 3 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S3405)。

【0170】

続いて、導出された CP_0, CP_2, CP_3 により 3 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S3406)、構築可能である場合 (ステップ

50

S 3 4 0 6 : Y E S)、C P 0、C P 2、C P 3 による 3 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S 3 4 0 7)。

【 0 1 7 1 】

続いて、導出された C P 1、C P 2、C P 3 により 3 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S 3 4 0 8)、構築可能である場合 (ステップ S 3 4 0 8 : Y E S)、C P 1、C P 2、C P 3 による 3 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S 3 4 0 9)。

【 0 1 7 2 】

続いて、導出された C P 0、C P 1 により 2 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S 3 4 1 0)、構築可能である場合 (ステップ S 3 4 1 0 : Y E S)、C P 0、C P 1 による 2 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S 3 4 1 1)。

【 0 1 7 3 】

続いて、導出された C P 0、C P 2 により 2 本制御点によるアフィンモデルを構築可能であるか否かを判断し (ステップ S 3 4 1 2)、構築可能である場合 (ステップ S 3 4 1 2 : Y E S)、C P 0、C P 2 による 2 本制御点アフィンモデルをアフィンマージ候補とする (ステップ S 3 4 1 3)。

【 0 1 7 4 】

ここで、アフィンモデルを構築可能であるか否かは、少なくとも、すべての制御点の参照画像が同一である (アフィン変換可能) ことを条件とする。また、C P 0、C P 1、C P 2 による 3 本制御点アフィンモデル、C P 0、C P 1 による 2 本制御点アフィンモデル以外のアフィンモデルは、3 本制御点アフィンモデルについては、C P 0、C P 1、C P 2 による 3 本制御点アフィンモデルに、2 本制御点アフィンモデルについては、C P 0、C P 1 による 2 本制御点アフィンモデルに変換する。

【 0 1 7 5 】

< アフィン固定マージ候補導出 >

アフィン固定マージ候補導出部 3 8 5 について説明する。アフィン固定マージ候補導出部 4 8 5 についてもアフィン固定マージ候補導出部 3 8 5 と同様である。

【 0 1 7 6 】

アフィン固定マージ候補は、固定された動き情報で制御点の動き情報を固定する。

【 0 1 7 7 】

具体的には、各制御点の動きベクトルを (0 , 0) に固定する。

【 0 1 7 8 】

< 時間予測動きベクトル導出 >

時間予測動きベクトルの説明に先行して、ピクチャの時間的な前後関係について図 4 9 を参照して説明する。図 4 9 (a) は、処理対象の符号化ブロックと、処理対象ピクチャとは時間的に異なる符号化済みのピクチャの関係を示す。処理対象ピクチャの符号化において参照する、特定の符号化済みのピクチャを ColPic と定義する。ColPic はシンタックスにより特定される。

【 0 1 7 9 】

また、図 4 9 (b) は、ColPic において、処理対象の符号化ブロックと同一位置、およびその近傍に存在する、符号化済みの符号化ブロックを示す。ただし、図 4 9 (b) に示した T 0 および T 1 の符号化ブロックは模式的なものであり、実際の位置や大きさはこの限りでない。いま、処理対象の符号化ブロックについて、位置を (xCb, yCb)、幅を cbWidth、高さを cbHeight とする。そして、

$$xColBr = xCb + cbWidth$$

$$yColBr = yCb + cbHeight$$

を算出する。位置 ((xCb + 3), (yCb + 3)) を含む ColPic 上の符号化ブロックが T 0 となる。また、

$$xColCtr = xCb + (cbWidth - 1)$$

10

20

30

40

50

$yColCtr = yCb + (cbHeight - 1)$
 を算出する。位置 $((xColCtr - 3) - 3, (yColCtr - 3) - 3)$ を含む ColPic 上の符号化ブロックが T 1 となる。

【 0 1 8 0 】

上記したピクチャの時間的な前後関係の説明は、符号化時のものであるが、復号時も同様となる。つまり、復号時は、上記の説明における符号化を復号と置き換えて、同様に説明される。

【 0 1 8 1 】

図 1 7 の通常予測動きベクトルモード導出部 3 0 1 における時間予測動きベクトル候補導出部 3 2 2 の動作について、図 5 0 を参照して説明する。

10

【 0 1 8 2 】

まず、ColPic を導出する (ステップ S 4 2 0 1)。ColPic の導出について、図 5 1 を参照して説明する。

【 0 1 8 3 】

スライスタイプ slice_type が B スライスで、フラグ collocated_from_l0_flag が 0 の場合 (ステップ S 4 2 1 1 : YES、ステップ S 4 2 1 2 : YES)、異なる時間のピクチャ ColPic は、参照リスト L 1 の参照インデックスが 0 のピクチャ RefPicList1[0] となる (ステップ S 4 2 1 3)。そうでない場合、すなわちスライスタイプ slice_type が B スライスで前述のフラグ collocated_from_l0_flag が 1 の場合 (ステップ S 4 2 1 1 : YES、ステップ S 4 2 1 2 : NO)、またはスライスタイプ slice_type が P スライスの場合 (ステップ S 4 2 1 1 : NO、ステップ S 4 2 1 4 : YES)、異なる時間のピクチャ ColPic は、参照リスト L 0 の参照インデックスが 0 のピクチャ RefPicList0[0] となる (ステップ S 4 2 1 5)。slice_type が P スライスでない場合 (ステップ S 4 2 1 4 : NO)、処理を終了する。

20

【 0 1 8 4 】

再び、図 5 0 を参照する。ColPic を導出したら、符号化ブロック colCb を導出し、符号化情報を取得する (ステップ S 4 2 0 2)。この処理について、図 5 2 を参照して説明する。

【 0 1 8 5 】

まず、異なる時間のピクチャ ColPic 内で、符号化対象の符号化ブロックと同一位置の右下位置を含む符号化ブロックを、異なる時間の符号化ブロック colCb とする (ステップ S 4 2 2 1)。この符号化ブロックの例を、図 4 9 の符号化ブロック T 0 に示す。

30

【 0 1 8 6 】

次に、異なる時間の符号化ブロック colCb の符号化情報を取得する (ステップ S 4 2 2 2)。異なる時間の符号化ブロック colCb の PredMode が利用できないか、異なる時間の符号化ブロック colCb の予測モード PredMode がイントラ予測 (MODE_INTRA) である場合 (ステップ S 4 2 2 3 : NO、ステップ S 4 2 2 4 : YES)、異なる時間のピクチャ ColPic 内で処理対象の符号化ブロックと同一位置の中央右下位置を含む符号化ブロックを異なる時間の符号化ブロック colCb とする (ステップ S 4 2 2 5)。この符号化ブロックの例を、図 4 9 の符号化ブロック T 1 に示す。

40

【 0 1 8 7 】

再び、図 5 0 を参照する。次に、参照リストごとに、インター予測情報を導出する (ステップ S 4 2 0 3、S 4 2 0 4)。ここでは、符号化ブロック colCb について、参照リストごとの動きベクトル mvLXCol と符号化情報が有効か否かを示すフラグ availableFlagLXCol を導出する。LX は参照リストを示し、参照リスト 0 の導出では LX は L0 となり、参照リスト 1 の導出では LX は L1 となる。インター予測情報の導出について、図 5 3 を参照して説明する。

【 0 1 8 8 】

異なる時間の符号化ブロック colCb が利用できない場合 (ステップ S 4 2 3 1 : NO)、または予測モード PredMode がイントラ予測 (MODE_INTRA) の場合 (ステップ S 4

50

2 3 2 : N O)、フラグavailableFlagLXColとフラグpredFlagLXColを共に 0 とし (ステップ S 4 2 3 3)、動きベクトルmvLXColを (0 , 0) として (ステップ S 4 2 3 4)、処理を終了する。

【 0 1 8 9 】

符号化ブロックcolCbが利用でき (ステップ S 4 2 3 1 : Y E S)、予測モードPredModeがイントラ予測 (MODE_INTRA) でない場合 (ステップ S 4 2 3 2 : Y E S)、以下の手順でmvCol、refIdxColおよびavailableFlagColを算出する。

【 0 1 9 0 】

符号化ブロックcolCbの L 0 予測が利用されているか否かを示すフラグPredFlagL0[xPCol][yPCol]が 0 の場合 (ステップ S 4 2 3 5 : Y E S)、符号化ブロックcolCbの予測モードはPred_L1であるので、動きベクトルmvColが符号化ブロックcolCbの L 1 の動きベクトルであるMvL1[xPCol][yPCol]と同じ値に設定され (ステップ S 4 2 3 6)、参照インデックスrefIdxColが L 1 の参照インデックスRefIdxL1[xPCol][yPCol]と同じ値に設定され (ステップ S 4 2 3 7)、参照リストlistColが L 1 に設定される (ステップ S 4 2 3 8)。ここで、xPCol、yPColは異なる時間のピクチャColPic内での符号化ブロックcolCbの左上の画素の位置を示すインデックスである。

【 0 1 9 1 】

一方、符号化ブロックcolCbの L 0 予測フラグPredFlagL0[xPCol][yPCol]が 0 でない場合 (ステップ S 4 2 3 5 : N O)、符号化ブロックcolCbの L 1 予測フラグPredFlagL1[xPCol][yPCol]が 0 か否かを判定する。符号化ブロックcolCbの L 1 予測フラグPredFlagL1[xPCol][yPCol]が 0 の場合 (ステップ S 4 2 3 9 : Y E S)、動きベクトルmvColが符号化ブロックcolCbの L 0 の動きベクトルであるMvL0[xPCol][yPCol]と同じ値に設定され (ステップ S 4 2 4 0)、参照インデックスrefIdxColが L 0 の参照インデックスRefIdxL0[xPCol][yPCol]と同じ値に設定され (ステップ S 4 2 4 1)、参照リストlistColが L 0 に設定される (ステップ S 4 2 4 2)。

【 0 1 9 2 】

符号化ブロックcolCbの L 0 予測フラグPredFlagL0[xPCol][yPCol]と符号化ブロックcolCbの L 1 予測フラグPredFlagL1[xPCol][yPCol]が共に 0 でない場合 (ステップ S 4 2 3 5 : N O、かつ S 4 2 3 9 : N O)、符号化ブロックcolCbのインター予測モードは双予測 (Pred_BI) であるので、L 0、L 1 の 2 つの動きベクトルから、一方を選択する (ステップ S 4 2 4 3)。

【 0 1 9 3 】

図 5 4 は、符号化ブロックcolCbのインター予測モードが双予測 (Pred_BI) のときの符号化ブロックのインター予測情報の導出処理手順を示すフローチャートである。

【 0 1 9 4 】

まず、すべての参照リストに登録されているすべてのピクチャの P O C が現在の処理対象ピクチャの P O C より小さいか否かを判定し (ステップ S 4 2 5 1)、符号化ブロックcolCbのすべての参照リストである L 0 及び L 1 に登録されているすべてのピクチャの P O C が現在の処理対象ピクチャの P O C より小さい場合で (ステップ S 4 2 5 1 : Y E S)、L X が L 0、即ち処理対象の符号化ブロックの L 0 の動きベクトルの予測ベクトル候補を導出している場合 (ステップ S 4 2 5 2 : Y E S)、符号化ブロックcolCbの L 0 の方のインター予測情報を選択し、L X が L 1、即ち処理対象の符号化ブロックの L 1 の動きベクトルの予測ベクトル候補を導出している場合 (ステップ S 4 2 5 2 : N O)、符号化ブロックcolCbの L 1 の方のインター予測情報を選択する。一方、符号化ブロックcolCbのすべての参照リスト L 0 及び L 1 に登録されているピクチャの P O C の少なくとも 1 つが現在の処理対象ピクチャの P O C より大きい場合で (ステップ S 4 2 5 1 : N O)、フラグcollocated_from_l0_flagが 0 場合 (ステップ S 4 2 5 3 : Y E S)、符号化ブロックcolCbの L 0 の方のインター予測情報を選択し、フラグcollocated_from_l0_flagが 1 の場合 (ステップ S 4 2 5 3 : N O)、符号化ブロックcolCbの L 1 の方のインター予測情報を選択する。

【 0 1 9 5 】

符号化ブロックcolCbのL 0の方のインター予測情報を選択する場合（ステップS 4 2 5 2：YES、またはステップS 4 2 5 3：YES）、動きベクトルmvColがMvL0[xPCol][yPCol]と同じ値に設定され（ステップS 4 2 5 4）、参照インデックスrefIdxColがRefIdxL0[xPCol][yPCol]と同じ値に設定され（ステップS 4 2 5 5）、リストlistColがL 0に設定される（ステップS 4 2 5 6）。

【 0 1 9 6 】

符号化ブロックcolCbのL 1の方のインター予測情報を選択する場合（ステップS 4 2 5 2：NO、またはステップS 4 2 5 3：NO）、動きベクトルmvColがMvL1[xPCol][yPCol]と同じ値に設定され（ステップS 4 2 5 7）、参照インデックスrefIdxColがRefIdxL1[xPCol][yPCol]と同じ値に設定され（ステップS 4 2 5 8）、リストlistColがL 1に設定される（ステップS 4 2 5 9）。

10

【 0 1 9 7 】

図5 3に戻り、符号化ブロックcolCbからインター予測情報が取得できたら、フラグavailableFlagLXColとフラグpredFlagLXColを共に1とする（ステップS 4 2 4 4）。

【 0 1 9 8 】

続いて、動きベクトルmvColをスケーリングして、動きベクトルmvLXColとする（ステップS 4 2 4 5）。この動きベクトルmvLXColのスケーリング演算処理手順を図5 5を用いて説明する。

【 0 1 9 9 】

20

異なる時間のピクチャColPicのPOCから、符号化ブロックcolCbのリストlistColで参照する参照インデックスrefIdxColに対応する参照ピクチャのPOCを減算してピクチャ間距離tdを、

$$td = [\text{異なる時間のピクチャColPicのPOC}] - [\text{符号化ブロックcolCbのリストlistColで参照する参照ピクチャのPOC}]$$

と算出する（ステップS 4 2 6 1）。なお、異なる時間のピクチャColPicよりも符号化ブロックcolCbのリストlistColで参照する参照ピクチャのPOCの方が表示順序で前の場合、ピクチャ間距離tdは正の値となり、異なる時間のピクチャColPicよりも符号化ブロックcolCbのリストlistColで参照する参照ピクチャのPOCの方が表示順序で後の場合、ピクチャ間距離tdは負の値となる。

30

【 0 2 0 0 】

次に、現在の処理対象ピクチャのPOCから現在の処理対象ピクチャのリストLXが参照する参照ピクチャのPOCを減算してピクチャ間距離tbを、

$$tb = [\text{現在の処理対象ピクチャのPOC}] - [\text{時間マージ候補のLXの参照インデックスに対応する参照ピクチャのPOC}]$$

と算出する（ステップS 4 2 6 2）。なお、現在の処理対象ピクチャよりも現在の処理対象ピクチャのリストLXで参照する参照ピクチャの方が表示順序で前の場合、ピクチャ間距離tbは正の値となり、現在の処理対象ピクチャのリストLXで参照する参照ピクチャの方が表示順序で後の場合、ピクチャ間距離tbは負の値となる。

【 0 2 0 1 】

40

続いて、ピクチャ間距離tdとtbを比較し（ステップS 4 2 6 3）、ピクチャ間距離tdとtbが等しい場合（ステップS 4 2 6 3：YES）、動きベクトルmvLXColを、

$$mvLXCol = mvCol$$

と算出して（ステップS 4 2 6 4）、本スケーリング演算処理を終了する。

【 0 2 0 2 】

一方、ピクチャ間距離tdとtbが等しくない場合（ステップS 4 2 6 3：NO）、変数txを、

$$tx = (16384 + \text{Abs}(td) \times 1) / td$$

と算出する（ステップS 4 2 6 5）。続いて、スケーリング係数distScaleFactorを、

$$\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \times 6)$$

50

と算出する（ステップS 4 2 6 6）。ここで、Clip3(x,y,z)は値zについて、最小値をx、最大値をyに制限する関数である。続いて、動きベクトルmvLXColを、

$$\text{mvLXCol} = \text{Clip3}(-32768, 32767, \text{Sign}(\text{distScaleFactor} * \text{mvLXCol}) * ((\text{Abs}(\text{distScaleFactor} * \text{mvLXCol}) + 127) \gg 8))$$

と算出して（ステップS 4 2 6 7）、本スケーリング演算処理を終了する。ここで、Sign(x)は値xの符号を返す関数であり、Abs(x)は値xの絶対値を返す関数である。

【0 2 0 3】

再び、図50を参照する。そして、L0の動きベクトルmvL0Colとを前述の通常予測動きベクトルモード導出部301における予測動きベクトル候補リストmvpListLXに候補として追加する（ステップS 4 2 0 5）。ただし、この追加は、参照リスト0の符号化ブロックcolCbが有効か否かを示すフラグavailableFlagL0Col=1の場合のみである。また、L1の動きベクトルmvL1Colを前述の通常予測動きベクトルモード導出部301における予測動きベクトル候補リストmvpListLXに候補として追加する（ステップS 4 2 0 5）。ただし、この追加は、参照リスト1の符号化ブロックcolCbが有効か否かを示すフラグavailableFlagL1Col=1の場合のみである。以上により、時間予測動きベクトル候補導出部322の処理を終了する。

10

【0 2 0 4】

上記した通常予測動きベクトルモード導出部301の説明は、符号化時のものであるが、復号時も同様となる。つまり、図23の通常予測動きベクトルモード導出部401における時間予測動きベクトル候補導出部422の動作は、上記の説明における符号化を復号と置き換えて、同様に説明される。

20

【0 2 0 5】

<時間マージ候補導出>

図18の通常マージモード導出部302における時間マージ候補導出部342の動作について、図56を参照して説明する。

【0 2 0 6】

まず、ColPicを導出する（ステップS 4 3 0 1）。次に、符号化ブロックcolCbを導出し、符号化情報を取得する（ステップS 4 3 0 2）。さらに、参照リストごとに、インター予測情報を導出する（ステップS 4 3 0 3, S 4 3 0 4）。以上の処理は、時間予測動きベクトル候補導出部322におけるS 4 2 0 1からS 4 2 0 4と同じであるため、説明を省略する。

30

【0 2 0 7】

次に、符号化ブロックcolCbが有効か否かを示すフラグavailableFlagColを算出する（ステップS 4 3 0 5）。フラグavailableFlagL0Col、またはフラグavailableFlagL1Colが1の場合に、availableFlagColは1となる。それ以外ではavailableFlagColは0となる。

【0 2 0 8】

そして、L0の動きベクトルmvL0Col、およびL1の動きベクトルmvL1Colを、前述の通常マージモード導出部302におけるマージ候補リストmergeCandListに候補として追加する（ステップS 4 3 0 6）。ただし、この追加は、符号化ブロックcolCbが有効か否かを示すフラグavailableFlagCol=1の場合のみである。以上により、時間マージ候補導出部342の処理を終了する。

40

【0 2 0 9】

上記した時間マージ候補導出部342の説明は、符号化時のものであるが、復号時も同様となる。つまり、図24の通常マージモード導出部402における時間マージ候補導出部442の動作は、上記の説明における符号化を復号と置き換えて、同様に説明される。

【0 2 1 0】

<履歴予測動きベクトル候補リストの更新>

次に、符号化側の符号化情報格納メモリ111及び復号側の符号化情報格納メモリ205に備える履歴予測動きベクトル候補リストHmvpCandListの初期化と更新方法について

50

詳細に説明する。図 3 8 は履歴予測動きベクトル候補リスト初期化・更新処理手順を説明するフローチャートである。

【 0 2 1 1 】

本実施の形態では、履歴予測動きベクトル候補リストHmvpCandListの更新は、符号化情報格納メモリ 1 1 1 及び符号化情報格納メモリ 2 0 5 で実施されるものとする。インター予測部 1 0 2 及びインター予測部 2 0 3 の中に履歴候補リスト更新部を設置して履歴予測動きベクトル候補リストHmvpCandListの更新を実施させてもよい。

【 0 2 1 2 】

スライスの先頭で履歴予測動きベクトル候補リストHmvpCandListの初期設定を行い、符号化側では予測方法決定部 1 0 5 で通常予測ベクトルモードまたは通常マージモードが選択された場合に履歴予測動きベクトル候補リストHmvpCandListを更新し、復号側では、ビット列復号部 2 0 1 で復号されたインター予測情報が通常予測ベクトルモードまたは通常マージモードの場合に履歴予測動きベクトル候補リストHmvpCandListを更新する。

【 0 2 1 3 】

通常予測ベクトルモードまたは通常マージモードでインター予測を行う際に用いるインター予測情報を、インター予測情報候補hMvpCandとして履歴予測動きベクトル候補リストHmvpCandListに登録する。インター予測情報候補hMvpCandには、L 0 の参照インデックスrefIdxL0及びL 1 の参照インデックスrefIdxL1、L 0 予測が行われるか否かを示すL 0 予測フラグpredFlagL0およびL 1 予測が行われるか否かを示すL 1 予測フラグpredFlagL1、L 0 の動きベクトルmvL0、L 1 の動きベクトルmvL1が含まれる。符号化側の符号化情報格納メモリ 1 1 1 及び復号側の符号化情報格納メモリ 2 0 5 に備える履歴予測動きベクトル候補リストHmvpCandListに登録されている要素（すなわち、インター予測情報）の中に、インター予測情報候補hMvpCandと同じ値のインター予測情報が存在するかどうかを履歴予測動きベクトル候補リストHmvpCandListの先頭から後方に向かって順に確認する。インター予測情報候補hMvpCandと同じ値のインター予測情報が存在する場合は、履歴予測動きベクトル候補リストHmvpCandListからその要素を削除する。一方、インター予測情報候補hMvpCandと同じ値のインター予測情報が存在しない場合は、履歴予測動きベクトル候補リストHmvpCandListの先頭の要素を削除し、履歴予測動きベクトル候補リストHmvpCandListの最後に、インター予測情報候補hMvpCandを追加する。

【 0 2 1 4 】

本発明の符号化側の符号化情報格納メモリ 1 1 1 及び復号側の符号化情報格納メモリ 2 0 5 に備える最大の数である最大履歴予測動きベクトル候補リストのサイズ、すなわち履歴予測動きベクトル候補リストHmvpCandListの要素の最大数（最大候補数）MaxNumHmvpCandは 6 とする。なお、MaxNumHmvpCandは、最大マージ候補数MaxNumMergeCand - 1と同じ値としてもよいし、最大マージ候補数MaxNumMergeCandと同じ値としてもよいし、5、6等所定の固定値でもよい。

【 0 2 1 5 】

まず、スライス単位での履歴予測動きベクトル候補リストHmvpCandListの初期化を行う（図 3 8 のステップ S 2 1 0 1 ）。スライスの先頭で履歴予測動きベクトル候補リストHmvpCandListのすべての要素を空にし、履歴予測動きベクトル候補リストHmvpCandListに登録されている履歴予測動きベクトル候補の数（現在の候補数）NumHmvpCandの値は 0 に設定する。

【 0 2 1 6 】

さらに、オフセット値hMvpIdxOffsetに所定の値を設定する。オフセット値hMvpIdxOffsetに設定する値は 0 から（履歴予測動きベクトル候補リストのサイズMaxNumHmvpCand - 1）までのいずれかの所定値を設定する。オフセット値hMvpIdxOffsetは所定の値とするが、オフセット値hMvpIdxOffsetの値をシーケンス単位で符号化／復号することにより設定してもよいし、スライス単位で符号化／復号することにより設定してもよい。オフセット値hMvpIdxOffsetについては詳細に後述する。

【 0 2 1 7 】

なお、履歴予測動きベクトル候補リストHmvpCandListの初期化をスライス単位（スライスの最初の符号化ブロック）で実施するとしたが、ピクチャ単位、タイル単位やツリーブロック行単位で実施しても良い。

【 0 2 1 8 】

続いて、スライス内の符号化ブロック毎に以下の履歴予測動きベクトル候補リストHmvpCandListの更新処理を繰り返し行なう（図38のステップS2102～S2107）。

【 0 2 1 9 】

まず、符号化ブロック単位での初期設定を行う。同一候補が存在するか否かを示すフラグidenticalCandExistにFALSE（偽）の値を設定し、削除対象の候補を示す削除対象インデックスremoveIdxに0を設定する（図38のステップS2103）。

10

【 0 2 2 0 】

履歴予測動きベクトル候補リストHmvpCandListに登録対象のインター予測情報候補hMvpCandと同じ値のインター予測情報が存在するか否かを判定する（図38のステップS2104）。符号化側の予測方法決定部105で通常予測動きベクトルモードまたは通常マージモードと判定された場合、または復号側のビット列復号部201で通常予測動きベクトルモードまたは通常マージモードとして復号された場合、そのインター予測情報を登録対象のインター予測情報候補hMvpCandとする。符号化側の予測方法決定部105でイントラ予測モード、サブブロック予測動きベクトルモードまたはサブブロックマージモードと判定された場合、または復号側のビット列復号部201でイントラ予測モード、サブブロック予測動きベクトルモードまたはサブブロックマージモードとして復号された場合、履歴予測動きベクトル候補リストHmvpCandListの更新処理を行わず、登録対象のインター予測情報候補hMvpCandは存在しない。登録対象のインター予測情報候補hMvpCandが存在しない場合はステップS2105～S2106をスキップする（図38のステップS2104：NO）。登録対象のインター予測情報候補hMvpCandが存在する場合はステップS2105以下の処理を行う（図38のステップS2104：YES）。

20

【 0 2 2 1 】

続いて、履歴予測動きベクトル候補リストHmvpCandListの各要素の中に登録対象のインター予測情報候補hMvpCandと同じ値のインター予測情報、すなわち同一の要素が存在するか否かを判定する（図38のステップS2105）。図39はこの同一要素確認処理手順のフローチャートである。履歴予測動きベクトル候補の数NumHmvpCandの値が0の場合（図39のステップS2121：NO）、履歴予測動きベクトル候補リストHmvpCandListは空で、同一候補は存在しないので図39ステップS2122～S2125をスキップし、本同一要素確認処理手順を終了する。履歴予測動きベクトル候補の数NumHmvpCandの値が0より大きい場合（図39のステップS2121：YES）、登録対象のインター予測情報候補hMvpCandと同じ値のインター予測情報が存在するかどうかを履歴予測動きベクトル候補リストHmvpCandListの先頭から後方に向かって順に確認する。履歴予測動きベクトルインデックスhMvpIdxが0からNumHmvpCand-1まで、ステップS2123の処理を繰り返す（図39のステップS2122～S2125）。まず、履歴予測動きベクトル候補リストの0から数えてhMvpIdx番目の要素HmvpCandList[hMvpIdx]がインター予測情報候補hMvpCandと同一か否かを比較する（図39のステップS2123）。同一の場合（図39のステップS2123：YES）、同一候補が存在するか否かを示すフラグidenticalCandExistにTRUE（真）の値を設定し、削除対象の要素の位置を示す削除対象インデックスremoveIdxに現在の履歴予測動きベクトルインデックスhMvpIndexの値を設定し、本同一要素確認処理を終了する。同一でない場合（図39のステップS2123：NO）、同一候補が存在するか否かを示すフラグidenticalCandExistはFALSE（偽）のままであり、hMvpIdxを1インクリメントし、履歴予測動きベクトルインデックスhMvpIdxがNumHmvpCand-1以下であれば、ステップS2123以降の処理を行う（図39のステップS2122～S2125）。

30

40

【 0 2 2 2 】

50

再び図 3 8 のフローチャートに戻り、履歴予測動きベクトル候補リストHmvpCandListの要素のシフト及び追加処理を行う（図 3 8 のステップ S 2 1 0 6）。図 4 0 は図 3 8 のステップ S 2 1 0 6 の履歴予測動きベクトル候補リストHmvpCandListの要素シフト／追加処理手順のフローチャートである。まず、履歴予測動きベクトル候補リストHmvpCandListに格納されている要素を除いてから新たな要素を追加するか、要素を除かずに新たな要素を追加するかを判定する。具体的には、同一候補が存在するか否かを示すフラグidenticalCandExistがTRUE（真）または現在の候補数NumHmvpCandが最大候補数MaxNumHmvpCandに達しているか否かを比較する（図 4 0 のステップ S 2 1 4 1）。現在の候補数NumHmvpCandが最大候補数MaxNumHmvpCandと同じ値である場合、履歴予測動きベクトル候補リストHmvpCandListに要素が最大数追加されていることを示す。同一候補が存在するか否かを示すフラグidenticalCandExistがTRUE（真）またはNumHmvpCandがMaxNumHmvpCandと同じ値のいずれかの条件を満たす場合（図 4 0 のステップ S 2 1 4 1：YES）、履歴予測動きベクトル候補リストHmvpCandListに格納されている要素を削除してから新たな要素を追加する。具体的には、同一候補が存在するか否かを示すフラグidenticalCandExistがTRUE（真）の場合は、履歴予測動きベクトル候補リストHmvpCandListから同一候補を削除する。NumHmvpCandがMaxNumHmvpCandと同じ値である場合、履歴予測動きベクトル候補リストHmvpCandListから先頭の候補（要素）を削除する。インデックスiの初期値をremoveIdx+1の値に設定する。removeIdxは削除対象の候補を示す削除対象インデックスである。このインデックスiが初期値のremoveIdx+1からNumHmvpCand-1まで、ステップ S 2 1 4 3 の要素シフト処理を繰り返す。（図 4 0 のステップ S 2 1 4 2 ～ S 2 1 4 4）。HMVP_CandList[i - 1]にHMVP_CandList[i]の要素をコピーすることで要素を前方にシフトし（図 4 0 のステップ S 2 1 4 3）、iを1インクリメントする（図 4 0 のステップ S 2 1 4 2 ～ S 2 1 4 4）。インデックスiがNumHmvpCand-1となり、ステップ S 2 1 4 3 の要素シフト処理が完了したら、履歴予測動きベクトル候補リストの最後にインター予測情報候補hMvpCandを追加する（図 4 0 のステップ S 2 1 4 5）。ここで、履歴予測動きベクトル候補リストの最後とは、0 から数えて(NumHmvpCand-1)番目のHMVP_CandList[NumHmvpCand-1]である。以上で、本履歴予測動きベクトル候補リストHMVP_CandListの要素シフト・追加処理を終了する。一方、同一候補が存在するか否かを示すフラグidenticalCandExistがTRUE（真）およびNumHmvpCandがMaxNumHmvpCandと同じ値のいずれの条件も満たさない場合（図 4 0 のステップ S 2 1 4 1：NO）、すなわち同一候補が存在するか否かを示すフラグidenticalCandExistがFALSE（偽）およびNumHmvpCandがMaxNumHmvpCandより小さい場合、履歴予測動きベクトル候補リストHmvpCandListに格納されている要素を除かずに、履歴予測動きベクトル候補リストの最後の要素の次の位置にインター予測情報候補hMvpCandを追加する（図 4 0 のステップ S 2 1 4 6）。ここで、履歴予測動きベクトル候補リストの最後の要素の次の位置とは、0 から数えてNumHmvpCand番目のHMVP_CandList[NumHmvpCand]である。履歴予測動きベクトル候補リストに要素が追加されていない場合は0番目の位置となる。また、NumHmvpCandを1インクリメントして、本履歴予測動きベクトル候補リストHMVP_CandListの要素シフト／追加処理を終了する。

【 0 2 2 3 】

図 4 3 は履歴予測動きベクトルリストの更新処理の一例を説明する図である。履歴予測動きベクトル候補リストHMVP_CandListに6つの要素（インター予測情報）が登録されている際に、新たなインター予測情報を追加する場合、履歴予測動きベクトル候補リストHMVP_CandListの各要素と前方から新たなインター予測情報を比較して（図 4 3（a））、新たなインター予測情報が履歴予測動きベクトル候補リストHMVP_CandListの先頭から3番目の要素HMVP2と同じ値であれば、履歴予測動きベクトル候補リストHMVP_CandListから要素HMVP2を削除して後方の要素HMVP3～HMVP5を前方に1つつつシフト（コピー）し、履歴予測動きベクトル候補リストHMVP_CandListの最後に新たなインター予測情報を追加して（図 4 3（b））、履歴予測動きベクトル候補リストHMVP_CandListの更

新を完了する（図 4 3（c））。

【 0 2 2 4 】

＜履歴予測動きベクトル候補導出処理＞

次に、符号化側の通常予測動きベクトルモード導出部 3 0 1 の履歴予測動きベクトル候補導出部 3 2 3、復号側の通常予測動きベクトルモード導出部 4 0 1 の履歴予測動きベクトル候補導出部 4 2 3 で共通の処理である図 2 0 のステップ S 3 0 4 の処理手順である履歴予測動きベクトル候補リスト HMVP CandList からの履歴予測動きベクトル候補の導出方法について詳細に説明する。図 4 1 は履歴予測動きベクトル候補導出処理手順を説明するフローチャートである。

【 0 2 2 5 】

現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数（ここでは 2 とする）以上または履歴予測動きベクトル候補の数（履歴予測動きベクトルリストに登録されている要素の数）NumHmvpCand の値が 0 の場合（図 4 1 のステップ S 2 2 0 1：NO）、図 4 1 のステップ S 2 2 0 2 から S 2 2 1 0 の処理を省略し、履歴予測動きベクトル候補導出処理手順を終了する。現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数である 2 より小さい場合、かつ履歴予測動きベクトル候補の数 NumHmvpCand の値が 0 より大きい場合（図 4 1 のステップ S 2 2 0 1：YES）、図 4 1 のステップ S 2 2 0 2 から S 2 2 1 0 の処理を行う。

【 0 2 2 6 】

続いて、インデックス i が 1 から、所定の上限值である 4 と履歴予測動きベクトル候補の数 NumHmvpCand のいずれか小さい値まで、図 4 1 のステップ S 2 2 0 3 から S 2 2 0 9 の処理を繰り返す（図 4 1 のステップ S 2 2 0 2 ~ S 2 2 1 0）。現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数である 2 以上の場合（図 4 1 のステップ S 2 2 0 3：NO）、図 4 1 のステップ S 2 2 0 4 から S 2 2 1 0 の処理を省略し、本履歴予測動きベクトル候補導出処理手順を終了する。現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数である 2 より小さい場合（図 4 1 のステップ S 2 2 0 3：YES）、図 4 1 のステップ S 2 2 0 4 以降の処理を行う。

【 0 2 2 7 】

続いて、ステップ S 2 2 0 5 から S 2 2 0 8 までの処理を履歴予測動きベクトル候補リスト HmvpCandList の各要素の参照リスト LY が L 0 と L 1）についてそれぞれ行う（図 4 1 のステップ S 2 2 0 4 ~ S 2 2 0 9）。履歴予測動きベクトル候補リスト HmvpCandList の L 0 と L 1 についてそれぞれ図 4 1 のステップ S 2 2 0 5 ~ S 2 2 0 8 の処理を行うことを示す。現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数である 2 以上の場合（図 4 1 のステップ S 2 2 0 5：NO）、図 4 1 のステップ S 2 2 0 7 から S 2 2 1 0 の処理を省略し、本履歴予測動きベクトル候補導出処理手順を終了する。現在の予測動きベクトル候補の数 numCurrMvpCand が予測動きベクトル候補リスト mvpListLX の最大要素数である 2 より小さい場合（図 4 1 のステップ S 2 2 0 5：YES）、図 4 1 のステップ S 2 2 0 6 以降の処理を行う。

【 0 2 2 8 】

続いて、履歴予測動きベクトル候補リストの要素の動きベクトルを予測動きベクトル候補として、予測動きベクトル候補リストに追加する。この時、履歴予測動きベクトル候補リストの後ろからオフセット値 hMvpIdxOffset で指定した個数の要素について、予測動きベクトル候補リストに含まれていない要素かどうかを降順に確認して、予測動きベクトル候補リストに含まれていない要素を予測動きベクトル候補リストに追加していき、その後は降順に予測動きベクトル候補リストに含まれていない要素かどうかを確認せずに履歴予測動きベクトル候補リストの要素を予測動きベクトル候補リストに追加していく。履歴予測動きベクトル候補リストの確認の際に、履歴予測動きベクトル候補リストの後ろからオフセット値 hMvpIdxOffset で指定した個数の要素だけを比較する理由について、図 6 5 を

10

20

30

40

50

用いて説明する。図 6 5 はブロックを 4 分割した場合の 3 つの例と履歴予測動きベクトル候補リストとの関係を示す。各符号化ブロックが通常予測動きベクトルモード、または通常マージモードで符号化された場合について説明する。図 6 5 (a) は符号化 / 復号対象の符号化ブロックが右上のブロック場合の図である。この場合、符号化 / 復号対象の符号化ブロックの左側のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素 HMVP5 となる可能性が高い。図 6 5 (b) は符号化 / 復号対象の符号化ブロックが左下のブロックの場合の図である。この場合、符号化 / 復号対象の符号化ブロックの右上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素 HMVP5 となる可能性が高く、符号化 / 復号対象の符号化ブロックの上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から 2 番目の要素 HMVP4 となる可能性が高い。図 6 5 (c) は符号化 / 復号対象の符号化ブロックが右下のブロックの場合の図である。この場合、符号化 / 復号対象の符号化ブロックの左のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素 HMVP5 となる可能性が高く、符号化 / 復号対象の符号化ブロックの上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から 2 番目の要素 HMVP4 となる可能性が高く、符号化 / 復号対象の符号化ブロックの左上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から 3 番目の要素 HMVP3 となる可能性が高い。すなわち、履歴予測動きベクトル候補リストの最後の要素は空間予測動きベクトル候補として導出される可能性が最も高い。図 6 5 (d) に示すように空間予測動きベクトル候補として導出される可能性が最も高い履歴予測動きベクトル候補リストの最後の要素 HMVP5 のみを比較するために、オフセット値 $hMvpIdxOffset$ を 1 に設定する。さらに、空間予測動きベクトル候補として導出される可能性が 2 番目に高い履歴予測動きベクトル候補リストの最後から 2 番目の要素も比較するために、オフセット値 $hMvpIdxOffset$ を 2 に設定することもできる。さらに、空間予測動きベクトル候補として導出される可能性が 3 番目に高い履歴予測動きベクトル候補リストの最後から 3 番目の要素も比較するために、オフセット値 $hMvpIdxOffset$ を 3 に設定することもできる。このようにオフセット値 $hMvpIdxOffset$ に 1 以上の値を設定することで、履歴予測動きベクトル候補リストの要素を比較する最大回数が削減するために、最大処理量が削減する。なお、オフセット値 $hMvpIdxOffset$ に 0 を設定することで、履歴予測動きベクトル候補リストの要素の比較回数が 0 となり、比較処理が省略される。インデックス i がオフセット値 $hMvpIdxOffset$ より小さくない場合、すなわち予測動きベクトル候補リストに含まれていない要素かどうかを確認する場合 (図 4 1 のステップ S 2 2 0 6 : Y E S)、履歴予測動きベクトル候補リスト $HmvpCandList[NumHmvpCand - i]$ の LY の参照インデックスが、符号化 / 復号対象動きベクトルの参照インデックス $refIdxLX$ と同じ値であり、履歴予測動きベクトル候補リストの要素 $HmvpCandList[NumHmvpCand - i]$ の LY が、予測動きベクトル候補リスト $mvpListLX$ のどの要素とも異なる要素の場合 (図 4 1 のステップ S 2 2 0 7 : Y E S)、予測動きベクトル候補リストの最後の要素として、予測動きベクトル候補リストの 0 から数えて $numCurrMvpCand$ 番目の要素 $mvpListLX[numCurrMvpCand]$ に履歴予測動きベクトル候補 $HmvpCandList[NumHmvpCand - i]$ の LY の動きベクトルを予測動きベクトル候補リスト $mvpListLX$ に追加し (図 4 1 のステップ S 2 2 0 8)、現在の予測動きベクトル候補の数 $numCurrMvpCand$ を 1 インクリメントする。履歴予測動きベクトル候補リスト $HmvpCandList$ の中に、符号化 / 復号対象動きベクトルの参照インデックス $refIdxLX$ と同じ参照インデックスの要素であり、予測動きベクトルリスト $mvpListLX$ のどの要素とも異なる要素がない場合 (図 4 1 のステップ S 2 2 0 6 : N O)、ステップ S 2 2 0 8 の追加処理をスキップする。

【 0 2 2 9 】

一方、インデックス i がオフセット値 $hMvpIdxOffset$ より小さくない場合、すなわち予測動きベクトル候補リストに含まれていない要素かどうかを確認しない場合 (図 4 1 のステップ S 2 2 0 6 の N O)、予測動きベクトル候補リストの最後の要素として、予測動きベクトル候補リストの 0 から数えて $numCurrMvpCand$ 番目の要素 $mvpListLX[numCurrMvpCand]$ に履歴予測動きベクトル候補 $HmvpCandList[NumHmvpCand - i]$ の LY の動

10

20

30

40

50

きベクトルを追加し（図 4 1 のステップ S 2 2 0 8）、現在の予測動きベクトル候補の数 numCurrMvpCand を 1 インクリメントする。

【 0 2 3 0 】

以上の図 4 1 のステップ S 2 2 0 5 から S 2 2 0 8 の処理を L 0 と L 1 で双方ともに行う（図 4 1 のステップ S 2 2 0 4 ~ S 2 2 0 9）。

【 0 2 3 1 】

インデックス i を 1 インクリメントし（図 4 1 のステップ S 2 2 0 2、S 2 2 1 0）、インデックス i が所定の上限値である 4 と履歴予測動きベクトル候補の数 NumHmvpCand のいずれか小さい値以下の場合、再びステップ S 2 2 0 3 以降の処理を行う（図 4 1 のステップ S 2 2 0 2 ~ S 2 2 1 0）。

【 0 2 3 2 】

<履歴マージ候補導出処理>

次に、符号化側の通常マージモード導出部 3 0 2 の履歴マージ候補導出部 3 4 5、復号側の通常マージモード導出部 4 0 2 の履歴マージ候補導出部 4 4 5 で共通の処理である図 2 1 のステップ S 4 0 4 の処理手順である履歴マージ候補リスト HmvpCandList からの履歴マージ候補の導出方法について詳細に説明する。図 4 2 は履歴マージ候補導出処理手順を説明するフローチャートである。

【 0 2 3 3 】

まず、初期化処理を行う（図 4 2 のステップ S 2 3 0 1）。フラグ isPruned[i] の 0 から (numCurrMergeCand - 1) 番目のそれぞれの要素に FALSE の値を設定し、変数 numOrigMergeCand に現在のマージ候補リストに登録されている要素の数 numCurrMergeCand を設定する。

【 0 2 3 4 】

続いて、履歴予測動きベクトルリストの要素の中でマージ候補リストに含まれていない要素をマージ候補リストに追加する。この時、履歴予測動きベクトルリストの後ろから降順に確認し、追加していく。インデックス hMvpIdx の初期値を 1 に設定し、この初期値から NumHmvpCand まで、図 4 2 のステップ S 2 3 0 3 からステップ S 2 3 1 1 までの追加処理を繰り返す（図 4 2 のステップ S 2 3 0 2 ~ S 2 3 1 2）。現在のマージ候補リストに登録されている要素の数 numCurrMergeCand が (最大マージ候補数 MaxNumMergeCand - 1) 以下でなければ、すなわち現在のマージ候補リストに登録されている要素の数 numCurrMergeCand が最大マージ候補数 MaxNumMergeCand に達していれば、マージ候補リストのすべての要素にマージ候補が追加されたので、本履歴マージ候補導出処理を終了する（図 4 2 のステップ S 2 3 0 3 : NO）。現在のマージ候補リストに登録されている要素の数 numCurrMergeCand が (最大マージ候補数 MaxNumMergeCand - 1) 以下の場合（図 4 2 のステップ S 2 3 0 3 : YES）、ステップ S 2 3 0 4 以降の処理を行う。

【 0 2 3 5 】

同じ動き情報を示す変数 sameMotion に FALSE（偽）の値を設定する（図 4 2 のステップ S 2 3 0 4）。続いて、履歴予測動きベクトル候補リストの要素であるインター予測情報をマージ候補として、マージ候補リストに追加する。この時、履歴予測動きベクトル候補リストの後ろの要素（最近追加された要素）からオフセット値 hMvpIdxOffset で指定した個数の要素について、マージ候補リストに含まれていない要素かどうかを降順に確認して、マージ候補リストに含まれていない要素をマージ候補リストに追加していき、その後は降順にマージ候補リストに含まれていない要素かどうかを確認せずに履歴予測動きベクトル候補リストの要素をマージ候補リストに追加していく。履歴予測動きベクトル候補リストの確認の際に、履歴予測動きベクトル候補リストの後ろの要素（最近追加された要素）からオフセット値 hMvpIdxOffset で指定した個数の要素だけを比較する理由について、図 6 5 を用いて説明する。図 6 5 はブロックを 4 分割した場合の 3 つの例と履歴予測動きベクトル候補リストとの関係を示す。各符号化ブロックが通常予測動きベクトルモード、または通常マージモードで符号化された場合について説明する。図 6 5 (a) は符号化 / 復号対象の符号化ブロックが右上のブロック場合の図である。この場合、符号化 / 復号対

10

20

30

40

50

象の符号化ブロックの左側のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素HMVP5となる可能性が高い。図 6 5 (b) は符号化 / 復号対象の符号化ブロックが左下のブロックの場合の図である。この場合、符号化 / 復号対象の符号化ブロックの右上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素HMVP5となる可能性が高く、符号化 / 復号対象の符号化ブロックの上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から2番目の要素HMVP4となる可能性が高い。図 6 5 (c) は符号化 / 復号対象の符号化ブロックが右下のブロックの場合の図である。この場合、符号化 / 復号対象の符号化ブロックの左のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後の要素HMVP5となる可能性が高く、符号化 / 復号対象の符号化ブロックの上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から2番目の要素HMVP4となる可能性が高く、符号化 / 復号対象の符号化ブロックの左上のブロックのインター予測情報が履歴予測動きベクトル候補リストの最後から3番目の要素HMVP3となる可能性が高い。すなわち、履歴予測動きベクトル候補リストの最後の要素は空間マージ候補として導出される可能性が最も高い。図 6 5 (d) に示すように空間マージ候補として導出される可能性が最も高い履歴予測動きベクトル候補リストの最後の要素HMVP5のみを比較するために、オフセット値hMvpIdxOffsetを1に設定する。さらに、空間マージ候補として導出される可能性が2番目に高い履歴予測動きベクトル候補リストの最後から2番目の要素も比較するために、オフセット値hMvpIdxOffsetを2に設定することもできる。さらに、空間マージ候補として導出される可能性が3番目に高い履歴予測動きベクトル候補リストの最後から3番目の要素も比較するために、オフセット値hMvpIdxOffsetを3に設定することもできる。このようにオフセット値hMvpIdxOffsetに1から3の値を所定値として設定し、履歴予測動きベクトル候補リストの後ろの要素（最近追加された要素）からオフセット値hMvpIdxOffsetで指定した個数の要素と空間マージ候補、またはマージ候補リストに格納されている要素のみを比較対象とすることで、履歴予測動きベクトル候補リストの要素を比較する最大回数が削減するために、最大処理量が削減する。

【 0 2 3 6 】

インデックスhMvpIdxが所定のオフセット値hMvpIdxOffset以下の場合、すなわちマージ候補リストに含まれていない要素かどうかを確認する場合（図 4 2 のステップ S 2 3 0 5 の Y E S ）、インデックスiの初期値を0に設定し、この初期値からnumOrigMergeCand-1まで図 4 2 のステップ S 2 3 0 7 、 S 2 3 0 8 の処理を行う（図 4 2 の S 2 3 0 6 ~ S 2 3 0 9 ）。履歴予測動きベクトル候補リストの0から数えて(NumHmvpCand-hMvpIdx)番目の要素HmvpCandList[NumHmvpCand-hMvpIdx]がマージ候補リストの0から数えてi番目の要素mergeCandList[i]と同じ値かどうかを比較する（図 4 2 のステップ S 2 3 0 7 ）。マージ候補の同じ値とは、マージ候補が持つすべてのインター予測情報の構成要素（インター予測モード、L 0 及び L 1 の参照インデックス、L 0 及び L 1 の動きベクトル）の値が同じであることを示す。したがって、このステップ S 2 3 0 7 の比較処理では、isPruned[i]がFALSE（偽）の際に、mergeCandList[i]とHmvpCandList[NumHmvpCand-hMvpIdx]が持つすべての構成要素（インター予測モード、L 0 及び L 1 の参照インデックス、L 0 及び L 1 の動きベクトル）の値が同じかどうかを比較する。同じ値の場合（図 3 9 のステップ S 2 3 0 7 : Y E S ）、sameMotionおよびisPruned[i]共にTRUE（真）を設定する（図 4 2 のステップ S 2 3 0 8 ）。なお、フラグisPruned[i]はマージ候補リストの0から数えてi番目の要素が履歴予測動きベクトル候補リストのいずれかの要素と同じ値であることを示すフラグである。同じ値でない場合（図 3 9 のステップ S 2 3 0 7 : N O ）、ステップ S 2 3 0 8 の処理をスキップする。図 4 2 のステップ S 2 3 0 6 からステップ S 2 3 0 9 までの繰り返し処理が完了したらsameMotionがFALSE（偽）かどうかを比較し（図 4 2 のステップ S 2 3 1 0 ）、sameMotionがFALSE（偽）の場合（図 4 2 のステップ S 2 3 1 0 : Y E S ）、すなわち履歴予測動きベクトル候補リストの0から数えて(NumHmvpCand - hMvpIdx)番目の要素HmvpCandList[NumHmvpCand - hMvpIdx]はmergeCandListに存在しないので、マージ候補リストの最後

の要素として、マージ候補リストのnumCurrMergeCand番目のmergeCandList[numCurrMergeCand]に履歴予測動きベクトル候補リストの0から数えて(NumHmvpCand - hMvpIdx)番目の要素HmvpCandList[NumHmvpCand - hMvpIdx]を追加し、numCurrMergeCandを1インクリメントする(図42のステップS2311)。一方、インデックスhMvpIdxがオフセット値hMvpIdxOffsetより小さくない場合、すなわちマージ候補リストに含まれていない要素かどうかを確認しない場合(図42のステップS2305のNO)、マージ候補リストの最後の要素として、マージ候補リストのnumCurrMergeCand番目のmergeCandList[numCurrMergeCand]に履歴予測動きベクトル候補リストの0から数えて(NumHmvpCand - hMvpIdx)番目の要素HmvpCandList[NumHmvpCand - hMvpIdx]を追加し、numCurrMergeCandを1インクリメントする(図42のステップS2312)。

10

【0237】

さらに、インデックスhMvpIdxを1インクリメントし(図42のステップS2302)、図42のステップS2302～S2312の繰り返し処理を行う。

【0238】

履歴予測動きベクトル候補リストのすべての要素の確認が完了するか、マージ候補リストのすべての要素にマージ候補が追加されたら、本履歴マージ候補の導出処理を完了する。

【0239】

<平均マージ候補導出処理>

次に、符号化側の通常マージモード導出部302の平均マージ候補導出部344、復号側の通常マージモード導出部402の平均マージ候補導出部444で共通の処理である図21のステップS403の処理手順である平均マージ候補の導出方法について詳細に説明する。図62は平均マージ候補導出処理手順を説明するフローチャートである。

20

【0240】

まず、初期化処理を行う(図62のステップS1301)。変数numOrigMergeCandに現在のマージ候補リストに登録されている要素の数numCurrMergeCandを設定する。

【0241】

続いて、マージ候補リストの先頭から順に走査し、2つの動き情報を決定する。1つ目の動き情報を示すインデックスi=0、2つ目の動き情報を示すインデックスj=1とする。(図62のステップS1302～S1303)。現在のマージ候補リストに登録されている要素の数numCurrMergeCandが(最大マージ候補数MaxNumMergeCand-1)以下でなければ、マージ候補リストのすべての要素にマージ候補が追加されたので、本履歴マージ候補導出処理を終了する(図62のステップS1304)。現在のマージ候補リストに登録されている要素の数numCurrMergeCandが(最大マージ候補数MaxNumMergeCand-1)以下の場合は、ステップS1305以降の処理を行う。

30

【0242】

マージ候補リストのi番目の動き情報mergeCandList[i]とマージ候補リストのj番目の動き情報mergeCandList[j]がともに無効であるか否かを判定し(図62のステップS1305)、ともに無効である場合は、mergeCandList[i]とmergeCandList[j]の平均マージ候補の導出を行わず、次の要素に移る。mergeCandList[i]とmergeCandList[j]がともに無効でない場合は、Xを0と1として以下の処理を繰り返す(図62のステップS1306からS1314)。

40

【0243】

mergeCandList[i]のLX予測が有効であるかを判定する(図62のステップS1307)。mergeCandList[i]のLX予測が有効である場合は、mergeCandList[j]のLX予測が有効であるかを判定する(図62のステップS1308)。mergeCandList[j]のLX予測が有効である場合、すなわち、mergeCandList[i]のLX予測とmergeCandList[j]のLX予測がともに有効である場合は、mergeCandList[i]のLX予測の動きベクトルとmergeCandList[j]のLX予測の動きベクトルを平均したLX予測の動きベクトルとmergeCandList[i]のLX予測の参照インデックスを有するLX予測の平均マージ候補を導出してaverageCandの

50

LX予測に設定し、averageCandのLX予測を有効とする（図62のステップS1309）。図62のステップS1308で、mergeCandList[j]のLX予測が有効でない場合、すなわち、mergeCandList[i]のLX予測が有効、かつmergeCandList[j]のLX予測が無効である場合は、mergeCandList[i]のLX予測の動きベクトルと参照インデックスを有するLX予測の平均マージ候補を導出してaverageCandのLX予測に設定し、averageCandのLX予測を有効とする（図62のステップS1310）。図62のステップS1307で、mergeCandList[i]のLX予測が有効でない場合、mergeCandList[j]のLX予測が有効であるか否かを判定する（図62のステップS1311）。mergeCandList[j]のLX予測が有効である場合、すなわちmergeCandList[i]のLX予測が無効、かつmergeCandList[j]のLX予測が有効である場合は、mergeCandList[j]のLX予測の動きベクトルと参照インデックスを有するLX予測の平均マージ候補を導出してaverageCandのLX予測に設定し、averageCandのLX予測を有効とする（図62のステップS1312）。図62のステップS1311で、mergeCandList[j]のLX予測が有効でない場合、すなわちmergeCandList[i]のLX予測、mergeCandList[j]のLX予測がともに無効である場合は、averageCandのLX予測を無効とする（図62のステップS1312）。

10

【0244】

以上のように生成されたL0予測、L1予測またはBI予測の平均マージ候補averageCandを、マージ候補リストのnumCurrMergeCand番目のmergeCandList[numCurrMergeCand]に追加し、numCurrMergeCandを1インクリメントする（図62のステップS1315）。以上で、平均マージ候補の導出処理を完了する。

20

【0245】

なお、平均マージ候補は動きベクトルの水平成分と動きベクトルの垂直成分それぞれで平均される。

【0246】

<サブブロック時間マージ候補導出>

図16のサブブロックマージモード導出部304におけるサブブロック時間マージ候補導出部381の動作について、図44を参照して説明する。

【0247】

まず、符号化ブロックが8x8画素未満か否かを判定する（ステップS4002）。

【0248】

30

符号化ブロックが8x8画素未満の場合（ステップS4002：YES）、サブブロック時間マージ候補の存在を示すフラグavailableFlagSbCol=0に設定して（ステップS4003）、サブブロック時間マージ候補導出部の処理を終了する。ここで、シンタックスによりテンポラル動きベクトル予測が禁止されている場合、またはサブブロック時間マージが禁止されている場合には、符号化ブロックが8x8画素未満の場合（ステップS4002：YES）と同じ処理をする。

【0249】

一方、符号化ブロックが8x8画素以上の場合（ステップS4002：NO）、符号化ピクチャにおける符号化ブロックの隣接動き情報を導出する（ステップS4004）。

【0250】

40

符号化ブロックの隣接動き情報を導出する処理について、図45を参照して説明する。隣接動き情報を導出する処理は、前述の空間予測動きベクトル候補導出部321の処理と相似している。ただし、隣接ブロックの探索をする順番はA0,B0,B1,A1であり、B2は探索しない。まず、隣接ブロックn=A0として、符号化情報を取得する（ステップS4052）。符号化情報とは、隣接ブロックを利用できるか否かを示すフラグavailableFlagN、参照リストごとの参照インデックスrefIdxLXN、および動きベクトルmvLXNを示す。

【0251】

次に、隣接ブロックnが有効が無効かを判断する（ステップS4054）。隣接ブロックを利用できるか否かを示すフラグavailableFlagN=1であれば有効、それ以外は無効とする。

50

【 0 2 5 2 】

隣接ブロックnが有効であれば（ステップS 4 0 5 4：YES）、参照インデックスrefIdxLXNを、隣接ブロックnの参照インデックスrefIdxLXnとする（ステップS 4 0 5 6）。また、動きベクトルmvLXNを、隣接ブロックnの動きベクトルmvLXnとして（ステップS 4 0 5 6）、ブロックの隣接動き情報を導出する処理を終了する。

【 0 2 5 3 】

一方、隣接ブロックnが無効であれば（ステップS 4 0 5 4：NO）、隣接ブロックn=B0として、符号化情報を取得し（ステップS 4 0 5 2）、隣接ブロックnが有効か無効かを判断する（ステップS 4 0 5 4）。以下、同様の処理をして、B1,A1の順番にループする。隣接動き情報を導出する処理は、隣接ブロックが有効となるまでループし、全ての隣接ブロックA0,B0,B1,A1が無効であれば、ブロックの隣接動き情報を導出する処理を終了する。

10

【 0 2 5 4 】

再び、図 4 4 を参照する。隣接動き情報を導出したら（ステップS 4 0 0 4）、テンポラル動きベクトルを導出する（ステップS 4 0 0 6）。

【 0 2 5 5 】

テンポラル動きベクトルを導出する処理について、図 4 6 を参照して説明する。まず、テンポラル動きベクトルtempMv=(0,0)として初期化する（ステップS 4 0 6 2）。

【 0 2 5 6 】

次に、隣接動き情報が有効か無効かを判断する（ステップS 4 0 6 4）。隣接ブロックを利用できるか否かを示すフラグavailableFlagN=1であれば有効、それ以外は無効とする。隣接動き情報が無効の場合（ステップS 4 0 6 4：NO）、テンポラル動きベクトルを導出する処理を終了する。

20

【 0 2 5 7 】

一方、隣接動き情報が有効の場合（ステップS 4 0 6 4：YES）、隣接ブロックNにおいてL1予測を利用しているか否かを示すフラグpredFlagL1Nが1か否かを判断する（ステップS 4 0 6 6）。predFlagL1N=0の場合（ステップS 4 0 6 6：NO）、次の処理（ステップS 4 0 7 8）に進む。predFlagL1N=1の場合（ステップS 4 0 6 6：YES）、すべての参照リストに登録されているすべてのピクチャのPOCが、現在の処理対象ピクチャのPOC以下か否かを判断する（ステップS 4 0 6 8）。この判断が真の場合（ステップS 4 0 6 8：YES）、次の処理（ステップS 4 0 7 0）に進む。

30

【 0 2 5 8 】

スライスタイプslice_typeがBスライスで、フラグcollocated_from_l0_flagが0の場合（ステップS 4 0 7 0：YES、ステップかつS 4 0 7 2：YES）、ColPicと参照ピクチャRefPicList1[refIdxL1N]（参照リストL1の参照インデックスrefIdxL1Nのピクチャ）が同じか否かを判断する（ステップS 4 0 7 4）。この判断が真の場合（ステップS 4 0 7 4：YES）、テンポラル動きベクトルtempMv=mvL1Nとする（ステップS 4 0 7 6）。この判断が偽の場合（ステップS 4 0 7 4：NO）、次の処理（ステップS 4 0 7 8）に進む。スライスタイプslice_typeがBスライスでなく、フラグcollocated_from_l0_flagが0でない場合（ステップS 4 0 7 0：NO、またはステップS 4 0 7 2：NO）、次の処理（ステップS 4 0 7 8）に進む。

40

【 0 2 5 9 】

そして、隣接ブロックNにおいてL0予測を利用しているか否かを示すフラグpredFlagL0Nが1か否かを判断する（ステップS 4 0 7 8）。predFlagL0N=1の場合（ステップS 4 0 7 8：YES）、ColPicと参照ピクチャRefPicList0[refIdxL0N]（参照リストL0の参照インデックスrefIdxL0Nのピクチャ）が同じか否かを判断する（ステップS 4 0 8 0）。この判断が真の場合（ステップS 4 0 8 0：YES）、テンポラル動きベクトルtempMv=mvL0Nとする（ステップS 4 0 8 2）。この判断が偽の場合（ステップS 4 0 8 0：NO）、テンポラル動きベクトルを導出する処理を終了する。

【 0 2 6 0 】

50

再び、図 4 4 を参照する。次に、ColPicを導出する（ステップ S 4 0 1 6）。この処理は、時間予測動きベクトル候補導出部 3 2 2 における S 4 2 0 1 と同じであるから、説明を省略する。

【 0 2 6 1 】

そして、異なる時間の符号化ブロック colCb を設定する（ステップ S 4 0 1 7）。これは、異なる時間のピクチャ ColPic 内で処理対象の符号化ブロックと同一位置の中央右下に位置する符号化ブロックを、colCb として設定するものである。この符号化ブロックは図 4 9 の符号化ブロック T 1 に相当する。

【 0 2 6 2 】

次に、符号化ブロック colCb にテンポラル動きベクトル tempMv を加算した位置を、新たな colCb とする（ステップ S 4 0 1 8）。いま、符号化ブロック colCb の左上の位置を (xColCb, yColCb)、テンポラル動きベクトル tempMv を 1/16 画素精度で (tempMv[0], tempMv[1]) とする。そして、

$$xColCb = \text{Clip3}(xCtb, xCtb + CtbSizeY + 3, xColCb + (\text{tempMv}[0] \quad 4))$$

$$yColCb = \text{Clip3}(yCtb, yCtb + CtbSizeY - 1, yColCb + (\text{tempMv}[1] \quad 4))$$

を算出する。ここで、ツリーブロックの左上の位置は (xCtb, yCtb)、ツリーブロックの大きさは CtbSizeY とする。位置 ((xCbCb - 3) - 3, (yColCb - 3) - 3) を含む ColPic 上の符号化ブロックが、新たな colCb となる。上式に示すように、tempMv 加算後の位置は、tempMv 加算前に比べて大きくずれないように、ツリーブロックの大きさ程度の範囲に補正される。もしこの位置が画面外となった場合は、画面内に補正される。

【 0 2 6 3 】

そして、この符号化ブロック colCb の予測モード PredMode がインター予測 (MODE_INTER) か否かを判定する（ステップ S 4 0 2 0）。colCb の予測モードがインター予測でない場合（ステップ S 4 0 2 0 : NO）、サブブロック時間マージ候補の存在を示すフラグ availableFlagSbCol = 0 に設定して（ステップ S 4 0 0 3）、サブブロック時間マージ候補導出部の処理を終了する。

【 0 2 6 4 】

一方、colCb の予測モードがインター予測の場合（ステップ S 4 0 2 0 : YES）、参照リストごとにインター予測情報を導出する（ステップ S 4 0 2 2、S 4 0 2 3）。ここでは、colCb について、参照リストごとの中心動きベクトル ctrMvLX と、LX 予測を利用しているか否かを示すフラグ ctrPredFlagLX を導出する。LX は参照リストを示し、参照リスト 0 の導出では LX は L0 となり、参照リスト 1 の導出では LX は L1 となる。インター予測情報の導出について、図 4 7 を参照して説明する。

【 0 2 6 5 】

異なる時間の符号化ブロック colCb が利用できない場合（ステップ S 4 1 1 2 : NO）、または予測モード PredMode がイントラ予測 (MODE_INTRA) の場合（ステップ S 4 1 1 4 : NO）、フラグ availableFlagLXCol とフラグ predFlagLXCol を共に 0 とし（ステップ S 4 1 1 6）、動きベクトル mvCol を (0, 0) として（ステップ S 4 1 1 8）、インター予測情報の導出処理を終了する。

【 0 2 6 6 】

符号化ブロック colCb が利用でき（ステップ S 4 1 1 2 : YES）、予測モード PredMode がイントラ予測 (MODE_INTRA) でない場合（ステップ S 4 1 1 4 : YES）、以下の手順で mvCol、refIdxCol および availableFlagCol を算出する。

【 0 2 6 7 】

符号化ブロック colCb の LX 予測が利用されているか否かを示すフラグ PredFlagLX[xPCol][yPCol] が 1 の場合（ステップ S 4 1 2 0 : YES）、動きベクトル mvCol が符号化ブロック colCb の LX の動きベクトルである MvLX[xPCol][yPCol] と同じ値に設定され（ステップ S 4 1 2 2）、参照インデックス refIdxCol が LX の参照インデックス RefIdxLX[xPCol][yPCol] と同じ値に設定され（ステップ S 4 1 2 4）、リスト listCol が LX に設定される（ステップ S 4 1 2 6）。ここで、xPCol、yPCol は異なる時間のピクチャ ColPi

10

20

30

40

50

c内での符号化ブロックcolCbの左上の画素の位置を示すインデックスである。

【0268】

一方、符号化ブロックcolCbのLX予測が利用されているか否かを示すフラグPredFlagLX[xPCol][yPCol]が0の場合（ステップS4120：NO）、以下の処理をする。まず、すべての参照リストに登録されているすべてのピクチャのPOCが、現在の処理対象ピクチャのPOC以下か否かを判断する（ステップS4128）。かつ、colCbのLY予測が利用されているか否かを示すフラグPredFlagLY[xPCol][yPCol]が1か否かを判断する（ステップS4128）。ここで、LY予測とはLX予測とは異なる参照リストと定義する。つまり、LX=L0ではLY=L1、LX=L1ではLY=L0となる。

【0269】

この判断が真の場合（ステップS4128：YES）、動きベクトルmvColが符号化ブロックcolCbのLYの動きベクトルであるMvLY[xPCol][yPCol]と同じ値に設定され（ステップS4130）、参照インデックスrefIdxColがLYの参照インデックスRefIdxLY[xPCol][yPCol]と同じ値に設定され（ステップS4132）、リストlistColがLXに設定される（ステップS4134）。

【0270】

一方、この判断が偽の場合（ステップS4128：NO）、フラグavailableFlagLXColとフラグpredFlagLXColを共に0とし（ステップS4116）、動きベクトルmvColを（0,0）として（ステップS4118）、インター予測情報の導出処理を終了する。

【0271】

符号化ブロックcolCbからインター予測情報が取得できたら、フラグavailableFlagLXColとフラグpredFlagLXColを共に1とする（ステップS4136）。

【0272】

続いて、動きベクトルmvColをスケーリングして、動きベクトルmvLXColとする（ステップS4138）。この処理は、時間予測動きベクトル候補導出部322におけるS4245と同じであるから、説明を省略する。

【0273】

再び、図44を参照する。参照リストごとにインター予測情報を導出したら、算出された動きベクトルmvLXColを中心動きベクトルctrMvLX、算出されたフラグpredFlagLXColをフラグctrPredFlagLXとする（ステップS4022,ステップS4023）。

【0274】

そして、中心動きベクトルが有効か無効かを判断する（ステップS4024）。ctrPredFlagL0=0かつctrPredFlagL1=0であれば無効、それ以外は無効と判断する。中心動きベクトルが無効の場合（ステップS4024：NO）、サブブロック時間マージ候補の存在を示すフラグavailableFlagSbCol=0に設定して（ステップS4003）、サブブロック時間マージ候補導出部の処理を終了する。

【0275】

一方、中心動きベクトルが有効の場合（ステップS4024：YES）、サブブロック時間マージ候補の存在を示すフラグavailableFlagSbCol=1に設定して（ステップS4025）、サブブロック動き情報を導出する（ステップS4026）。この処理について、図48を参照して説明する。

【0276】

まず、符号化ブロックcolCbの幅cbWidthと高さcBheightから、幅方向のサブブロック数numSbXおよび高さ方向のサブブロック数numSbYを算出する（ステップS4152）。また、refIdxLXSbCol=0とする（ステップS4152）。この処理以降は、予測サブブロックcolSbの単位で繰り返し処理をする。この繰り返しは、高さ方向のインデックスySbIdxを0からnumSbYまで、幅方向のインデックスxSbIdxを0からnumSbXまで変更しながら処理をする。

【0277】

符号化ブロックcolCbの左上の位置を(xCb,yCb)とすると、予測サブブロックcolSbの左

10

20

30

40

50

上の位置(xSb,ySb)は、

$$xSb = xCb + xSbIdx * sbWidth$$

$$ySb = yCb + ySbIdx * sbHeight$$

と算出される。次に、予測サブブロックcolSbにテンポラル動きベクトルtempMvを加算した位置を、新たなcolSbとする(ステップS 4 1 5 4)。予測サブブロックcolSbの左上の位置を(xColSb, yColSb)、テンポラル動きベクトルtempMvを1/16画素精度で(tempMv[0], tempMv[1])とすると、新たなcolSbの左上の位置は、

$$xColSb = Clip3(xCtb, xCtb + CtbSizeY + 3, xSb + (tempMv[0] \quad 4))$$

$$yColSb = Clip3(yCtb, yCtb + CtbSizeY - 1, ySb + (tempMv[1] \quad 4))$$

となる。ここで、ツリーブロックの左上の位置は(xCtb, yCtb)、ツリーブロックの大きさはCtbSizeYとする。上式に示すように、tempMv加算後の位置は、tempMv加算前に比べて大きくずれないように、ツリーブロックの大きさ程度の範囲に補正される。もしこの位置が画面外となった場合は、画面内に補正される。

【0 2 7 8】

そして、参照リストごとにインター予測情報を導出する(ステップS 4 1 5 6, S 4 1 5 8)。ここでは、予測サブブロックcolSbについて、サブブロック単位で参照リストごとの動きベクトルmvLXSbColと、予測サブブロックが有効か否かを示すフラグavailableFlagLXSbColを導出する。LXは参照リストを示し、参照リスト0の導出ではLXはL0となり、参照リスト1の導出ではLXはL1となる。インター予測情報の導出は、図47のS 4 0 2 2, S 4 0 2 3と同じであるため、説明を省略する。

【0 2 7 9】

インター予測情報を導出後(ステップS 4 1 5 6, S 4 1 5 8)、予測サブブロックcolSbが有効か否かを判断する(ステップS 4 1 6 0)。availableFlagL0SbCol=0かつavailableFlagL1SbCol=0の場合はcolSbが無効、それ以外は有効と判断する。colSbが無効の場合(ステップS 4 1 6 0: NO)、動きベクトルmvLXSbColを、中心動きベクトルctrMvLXとする(ステップS 4 1 6 2)。さらに、LX予測を利用しているか否かを示すフラグpredFlagLXSbColを、中心動きベクトルにおけるフラグctrPredFlagLXとする(ステップS 4 1 6 2)。以上により、サブブロック動き情報の導出を終了する。

【0 2 8 0】

再び、図44を参照する。そして、L0の動きベクトルmvL0SbCol、およびL1の動きベクトルmvL1SbColを、前述のサブブロックマージモード導出部304におけるサブブロックマージ候補リストsubblockMergeCandListに候補として追加する(ステップS 4 0 2 8)。ただし、この追加は、サブブロック時間マージ候補の存在を示すフラグavailableSbCol=1の場合のみである。以上により、時間マージ候補導出部342の処理を終了する。

【0 2 8 1】

上記したサブブロック時間マージ候補導出部381の説明は、符号化時のものであるが、復号時も同様となる。つまり、図22のサブブロックマージモード導出部404におけるサブブロック時間マージ候補導出部481の動作は、上記の説明における符号化を復号と置き換えて、同様に説明される。

【0 2 8 2】

<動き補償予測処理>

動き補償予測部306は、符号化において現在予測処理の対象となっているブロックの位置およびサイズを取得する。また、動き補償予測部306は、インター予測情報をインター予測モード判定部305から取得する。取得したインター予測情報から参照インデックスおよび動きベクトルを導出し、復号画像メモリ内の参照インデックスで特定される参照ピクチャを、動きベクトルの分だけ予測ブロックの画像信号と同一位置より移動させた位置の画像信号を取得した後に予測信号を生成する。

【0 2 8 3】

インター予測におけるインター予測モードがL0予測やL1予測のような、単一の参照ピク

10

20

30

40

50

チャからの予測の場合には、1つの参照ピクチャから取得した予測信号を動き補償予測信号とし、インター予測モードがBI予測のような、予測モードが2つの参照ピクチャからの予測の場合には、2つの参照ピクチャから取得した予測信号を重みづけ平均したものを動き補償予測信号とし、動き補償予測信号を予測方法決定部に供給する。ここでは双予測の重みづけ平均の比率を1:1とするが、他の比率を用いて重みづけ平均を行っても良い。例えば、予測対象となっているピクチャと参照ピクチャとのピクチャ間隔に近いものほど重みづけの比率が大きくなるようにしてもよい。また、重みづけ比率の算出をピクチャ間隔の組み合わせと重みづけ比率との対応表を用いて行うようにしても良い。

【0284】

動き補償予測部406は、符号化側の動き補償予測部306と同様の機能をもつ。動き補償予測部406は、インター予測情報を、通常予測動きベクトルモード導出部401、通常マージモード導出部402、サブブロック予測動きベクトルモード導出部403、サブブロックマージモード導出部404から、スイッチ408を介して取得する。

【0285】

動き補償予測部406は、得られた動き補償予測信号を、復号画像信号重畳部207に供給する。

【0286】

<インター予測モードについて>

単一の参照ピクチャからの予測を行う処理を単予測と定義し、単予測の場合はL0予測またはL1予測という、参照リストL0、L1に登録された2つの参照ピクチャのいずれか一方を利用した予測を行う。L0予測およびL1予測は前方向予測（前方の参照画像を参照する予測）であっても後方向予測（後方の参照画像を参照する予測）であってもよい。図57～58は、L0予測（単予測）での動き補償予測を説明するための図である。

【0287】

図57はインター予測モードがL0予測であってL0の参照ピクチャ（RefL0Pic）が処理対象ピクチャ（CurPic）より前の時刻にある場合を示している。図58はL0予測であってL0の参照ピクチャが処理対象ピクチャより後の時刻にある場合を示している。同様に、図57および図58のL0予測の参照ピクチャをL1予測の参照ピクチャ（RefL1Pic）に置き換えて単予測を行うこともできる。

【0288】

2つの参照ピクチャからの予測を行う処理を双予測と定義し、双予測の場合はL0予測とL1予測の双方を利用して双予測と表現する。図59～61は、双予測での動き補償予測を説明するための図である。図59は双予測であってL0予測の参照ピクチャが処理対象ピクチャより前の時刻にあって、L1予測の参照ピクチャが処理対象ピクチャより後の時刻にある場合を示している。図60は双予測であってL0予測の参照ピクチャとL1予測の参照ピクチャが処理対象ピクチャより前の時刻にある場合を示している。図61は双予測であってL0予測の参照ピクチャとL1予測の参照ピクチャが処理対象ピクチャより後の時刻にある場合を示している。

【0289】

このように、L0/L1の予測種別と時間の関係は、L0が前方向予測（前方の参照画像を参照する予測）、L1が後方向予測（後方の参照画像を参照する予測）とは限定されずに用いることが可能である。また双予測の場合に、同一の参照ピクチャを用いてL0予測及びL1予測のそれぞれを行ってもよい。なお、動き補償予測を単予測で行うか双予測で行うかの判断は、例えばL0予測を利用するか否か及びL1予測を利用するか否かを示す情報（例えば、フラグ）に基づき判断される。

【0290】

<参照インデックスについて>

本発明の実施の形態では、動き補償予測の精度向上のために、動き補償予測において複数の参照ピクチャの中から最適な参照ピクチャを選択することを可能とする。そのため、動き補償予測で利用した参照ピクチャを参照インデックスとして利用するとともに、参照

10

20

30

40

50

インデックスを符号化ベクトルとともに符号化ストリーム中に符号化する。

【0291】

＜通常予測動きベクトルモードに基づく動き補償処理＞

動き補償予測部306は、図16の符号化側におけるインター予測部102でも示されるように、インター予測モード判定部305において、通常予測動きベクトルモード導出部301によるインター予測情報が選択された場合には、このインター予測情報をインター予測モード判定部305から取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、予測方法決定部105に供給される。

【0292】

同様に、動き補償予測部406は、図22の復号側におけるインター予測部203でも示されるように、復号の過程でスイッチ408が通常予測動きベクトルモード導出部401に接続された場合には、通常予測動きベクトルモード導出部401によるインター予測情報を取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、復号画像信号重畳部207に供給される。

【0293】

＜通常マージモードに基づく動き補償処理＞

動き補償予測部306は、図16の符号化側におけるインター予測部102でも示されるように、インター予測モード判定部305において、通常マージモード導出部302によるインター予測情報が選択された場合には、このインター予測情報をインター予測モード判定部305から取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、予測方法決定部105に供給される。

【0294】

同様に、動き補償予測部406は、図22の復号側におけるインター予測部203でも示されるように、復号の過程でスイッチ408が通常マージモード導出部402に接続された場合には、通常マージモード導出部402によるインター予測情報を取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、復号画像信号重畳部207に供給される。

【0295】

＜サブブロック予測動きベクトルモードに基づく動き補償処理＞

動き補償予測部306は、図16の符号化側におけるインター予測部102でも示されるように、インター予測モード判定部305において、サブブロック予測動きベクトルモード導出部303によるインター予測情報が選択された場合には、このインター予測情報をインター予測モード判定部305から取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、予測方法決定部105に供給される。

【0296】

同様に、動き補償予測部406は、図22の復号側におけるインター予測部203でも示されるように、復号の過程でスイッチ408がサブブロック予測動きベクトルモード導出部403に接続された場合には、サブブロック予測動きベクトルモード導出部403によるインター予測情報を取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、復号画像信号重畳部207に供給される。

【0297】

＜サブブロックマージモードに基づく動き補償処理＞

動き補償予測部306は、図16の符号化側におけるインター予測部102でも示されるように、インター予測モード判定部305において、サブブロックマージモード導出部

10

20

30

40

50

304 によるインター予測情報が選択された場合には、このインター予測情報をインター予測モード判定部305から取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、予測方法決定部105に供給される。

【0298】

同様に、動き補償予測部406は、図22の復号側におけるインター予測部203でも示されるように、復号の過程でスイッチ408がサブブロックマージモード導出部404に接続された場合には、サブブロックマージモード導出部404によるインター予測情報を取得し、現在処理対象となっているブロックのインター予測モード、参照インデックス、動きベクトルを導出し、動き補償予測信号を生成する。生成された動き補償予測信号は、復号画像信号重畳部207に供給される。

10

【0299】

<アフィンモードに基づく動き補償処理>

本実施の形態においてはアフィンモデルによる動き補償が利用できる。アフィンモデルによる動き補償は符号化ブロックの2～4個の角を制御点とし、制御点の動きベクトルからサブブロックの動きベクトルを導出し、サブブロック単位で動き補償を行う。

【0300】

以下のフラグは、符号化処理においてインター予測モード判定部305により決定されるインター予測の条件に基づいて以下のフラグに反映され、符号化ストリーム中に符号化される。復号処理においては、符号化ストリーム中の以下のフラグに基づいてアフィンモデルによる動き補償を行うか否かを特定する。

20

【0301】

sps_affine_enabled_flagは、インター予測において、アフィンモデルによる動き補償が利用できるか否かを表す。sps_affine_enabled_flagが0であれば、シーケンス単位でアフィンモデルによる動き補償ではないように抑制される。また、inter_affine_flagとcu_affine_type_flagは、符号化ビデオシーケンスのCU（符号化ユニット）シンタックスにおいて伝送されない。sps_affine_enabled_flagが1であれば、符号化ビデオシーケンスにおいてアフィンモデルによる動き補償を利用できる。

【0302】

sps_affine_type_flagは、インター予測において、6パラメータアフィンモデルによる動き補償が利用できるか否かを表す。6パラメータアフィンモデルは3つの制御点のそれぞれの動きベクトルの水平及び垂直成分の6つのパラメータからサブブロックの動きベクトルを導出し、サブブロック単位で動き補償を行うモードである。サブブロック単位で動きベクトルを導出するが、符号化ブロック単位で共通の参照インデックスを導出する。

30

【0303】

sps_affine_type_flagが0であれば、6パラメータアフィンモデルによる動き補償ではないように抑制される。また、cu_affine_type_flagは、符号化ビデオシーケンスのCUシンタックスにおいて伝送されない。sps_affine_type_flagが1であれば、符号化ビデオシーケンスにおいて6パラメータアフィンモデルによる動き補償を利用できる。

【0304】

sps_affine_type_flagが存在しない場合には、0であるものとする。

40

【0305】

PまたはBスライスを復号している場合、現在処理対象となっているCUにおいて、inter_affine_flagが1であれば、現在処理対象となっているCUの動き補償予測信号を生成するために、アフィンモデルによる動き補償が用いられる。

【0306】

inter_affine_flagが0であれば、現在処理対象となっているCUにアフィンモデルは用いられない。

【0307】

inter_affine_flagが存在しない場合には、0であるものとする。

50

【 0 3 0 8 】

PまたはBスライスを復号している場合、現在処理対象となっているCUにおいて、cu_affine_type_flagが1であれば、現在処理対象となっているCUの動き補償予測信号を生成するために、6パラメータアフィンモデルによる動き補償が用いられる。

【 0 3 0 9 】

cu_affine_type_flagが0であれば、現在処理対象となっているCUの動き補償予測信号を生成するために、4パラメータアフィンモデルによる動き補償が用いられる。4パラメータアフィンモデルは2つの制御点のそれぞれの動きベクトルの水平成分及び垂直成分の4つのパラメータからサブブロックの動きベクトルを導出し、サブブロック単位で動き補償を行うモードである。

10

【 0 3 1 0 】

< マージ差分動きベクトル (MMVD) >

マージ候補の上位2つ(マージ候補リスト内のマージインデックスが0および1のマージ候補)の動きベクトルに対し、差分動きベクトルを加算することができる。この差分動きベクトルを、マージ差分動きベクトルと呼ぶ。

【 0 3 1 1 】

符号化側のマージ候補選択部347においてマージ差分動きベクトルを加算する場合、マージ差分動きベクトルが加算された動きベクトルは、インター予測モード判定部305を介して動き補償予測部306に供給される。また、ビット列符号化部108は、マージ差分動きベクトルに関する情報を符号化する。マージ差分動きベクトルに関する情報とは、動きベクトルに加算する距離を示すインデックスmmvd_distance_idxと、動きベクトルを加算する方向を示すインデックスmmvd_direction_idxである。これらのインデックスは、図63(a)および図63(b)に示す表のように定義される。そして、マージ差分動きベクトルオフセットMmvdOffsetのx,y成分をそれぞれMmvdOffset[0], MmvdOffset[1]で表すと、

20

$$\text{MmvdOffset}[0] = (\text{MmvdDistance} \quad 2) * \text{MmvdSign}[0]$$

$$\text{MmvdOffset}[1] = (\text{MmvdDistance} \quad 2) * \text{MmvdSign}[1]$$

となる。マージ差分動きベクトルは、上式のマージ差分動きベクトルオフセットMmvdOffsetより導出される。マージ差分動きベクトルを導出する詳細は、以下の復号側の場合において説明する。

30

【 0 3 1 2 】

復号側において、マージ差分動きベクトルが存在する場合、ビット列復号部201に供給されるビットストリームからマージ差分動きベクトルに関する情報を分離し、マージ差分動きベクトルオフセットMmvdOffsetを導出する。また、マージ候補選択部447は、復号されたマージ差分動きベクトルオフセットから、マージ差分動きベクトルを導出する。このマージ差分動きベクトルを動きベクトルに加算してから、その動きベクトルを動き補償予測部406に供給する。

【 0 3 1 3 】

マージ候補選択部447におけるマージ差分動きベクトルmMvdLXの導出について、図64(a)のフローチャートを参照して説明する。まず、符号化ブロックのインター予測モードが双予測(PRED_BI)であるか否かを判定する(S4402)。双予測でない場合(S4402:No)、L0予測(PRED_L0)であるか否かを判定する(S4404)。L0予測の場合(S4404:Yes)、

40

$$\text{mMvdL0} = \text{MmvdOffset}$$

$$\text{mMvdL1} = 0$$

として(S4406)、マージ差分動きベクトルを導出する処理は終了する。L1予測の場合(S4404:No)、

$$\text{mMvdL0} = 0$$

$$\text{mMvdL1} = \text{MmvdOffset}$$

として(S4408)、マージ差分動きベクトルを導出する処理は終了する。

50

【 0 3 1 4 】

一方、双予測の場合 (S 4 4 0 2 : Yes)、処理対象ピクチャcurrPicと参照ピクチャのPOCの差を、参照リストごとに計算し、それぞれcurrPocDiffL0, currPocDiffL1とする (S 4 4 1 0)。ここで、picAとpicBのPOCの差DiffPicOrderCnt(picA, picB)は、

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = [\text{picAのPOC}] - [\text{picBのPOC}]$$

を示す。また、参照ピクチャRefPicList0[refIdxL0]は、参照リストL0の参照インデックスrefIdxL0が示すピクチャである。同様に、参照ピクチャRefPicList1[refIdxL1]は、参照リストL1の参照インデックスrefIdxL1が示すピクチャである。

【 0 3 1 5 】

次に、 $-\text{currPocDiffL0} * \text{currPocDiffL1} = 0$ が否かを判定する (ステップ S 4 4 1 2)。この判定が真の場合 (ステップ S 4 4 1 2 : Yes)、

$$\text{mMvdL0} = \text{MmvdOffset}$$

$$\text{mMvdL1} = -\text{MmvdOffset}$$

として (ステップ S 4 4 1 4)、マージ差分動きベクトルを導出する処理は終了する。一方、この判定が偽の場合 (ステップ S 4 4 1 2 : No)、

$$\text{mMvdL0} = \text{MmvdOffset}$$

$$\text{mMvdL1} = \text{MmvdOffset}$$

とする (ステップ S 4 4 1 6)。次に、参照リストL0とのPOCの差の絶対値が、参照リストL1とのPOCの差の絶対値以上か否かを判定する (ステップ S 4 4 1 8)。この判定が真の場合 (ステップ S 4 4 1 8 : Yes)、 $X=0$, $Y=1$ とし (ステップ S 4 4 2 0)、L1のマージ差分動きベクトルmMvdL1をスケーリングする (ステップ S 4 4 2 4)。ここで、mMvdLYは、 $Y=0$ の場合はmMvdL0、 $Y=1$ の場合はmMvdL1であることを示す。一方、この判定が偽の場合 (ステップ S 4 4 1 8 : No)、 $X=1$, $Y=0$ とし (ステップ S 4 4 2 2)、L0のマージ差分動きベクトルmMvdL0をスケーリングする (ステップ S 4 4 2 4)。マージ差分動きベクトルmMvdLYのスケーリングは、図 6 4 (b)のように、

$$\text{td} = \text{Clip3}(-128, 127, \text{currPocDiffLX})$$

$$\text{tb} = \text{Clip3}(-128, 127, \text{currPocDiffLY})$$

$$\text{tx} = (16384 + \text{Abs}(\text{td}) - 1) / \text{td}$$

$$\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * \text{tx} + 32) > 6)$$

$$\text{mMvdLY} = \text{Clip3}(-32768, 32767, \text{Sign}(\text{distScaleFactor} * \text{mMvdLY})$$

$$* ((\text{Abs}(\text{distScaleFactor} * \text{mMvdLY}) + 127) > 8))$$

として導出する。ここで、currPocDiffLXは、 $X=0$ の場合はcurrPocDiffL0、 $X=1$ の場合はcurrPocDiffL1であることを示す。同様に、currPocDiffLYは、 $Y=0$ の場合はcurrPocDiffL0、 $Y=1$ の場合はcurrPocDiffL1であることを示す。また、Clip3(x,y,z)は値zについて、最小値をx、最大値をyに制限する関数である。Sign(x)は値xの符号を返す関数であり、Abs(x)は値xの絶対値を返す関数である。以上により、マージ差分動きベクトルを導出する処理は終了する。

【 0 3 1 6 】

マージ差分動きベクトルは、サブブロックマージ候補の上位2つの動きベクトルに対して加算しても良い。この場合、動きベクトルに加算する距離を示すインデックスmmvd_distance_idxは、図 6 3 (c) に示す表のように定義される。サブブロックマージ候補選択部 3 8 6 の動作は、マージ候補選択部 3 4 7 と同じであるため、説明を省略する。また、サブブロックマージ候補選択部 4 8 6 の動作は、マージ候補選択部 4 4 7 と同じであるため、説明を省略する。

【 0 3 1 7 】

前述の通り、MmvdDistanceは、図 6 3 (a) や図 6 3 (c) に示す表のように定義される。これらの表は1/4画素精度で定義されているので、生成されるマージ差分動きベクトルは、小数画素精度を含むことがある。ただし、これらの表の画素精度が1であることを示すフラグをスライス単位で符号化/復号することにより、生成されるマージ差分動きベクトルが、小数画素精度を含まないように変更することができる。

10

20

30

40

50

【0318】

< 適応動きベクトル解像度 (AMVR) >

符号化ブロック単位で、差分動きベクトルの解像度を適応的に変更することができる。この解像度を、適応動きベクトル解像度と呼ぶ。

【0319】

通常予測動きベクトルモードに対して適応動きベクトル解像度を用いる場合について説明する。この場合、空間予測動きベクトル候補導出部321および421と、時間予測動きベクトル候補導出部322および422と、履歴予測動きベクトル候補導出部323および423において、導出された候補の動きベクトルは解像度に応じて丸められる。解像度は1/4, 1, 4画素精度から選択でき、解像度を変更しない場合は1/4画素精度となる。丸め処理は、処理対象の符号化ブロックにおける動きベクトルの解像度に合わせてなされる。つまり、導出された候補の動きベクトルmvXは、

$$\begin{aligned} \text{rightShift} &= \text{leftShift} = \text{MvShift} + 2 \\ \text{offset} &= 1 \quad (\text{rightShift} - 1) \\ \text{mvX} &= (\text{mvX} = 0 ? (\text{mvX} + \text{offset}) \quad \text{rightShift} : \\ &\quad - ((-\text{mvX} + \text{offset}) \quad \text{rightShift})) \quad \text{leftShift} \end{aligned}$$

と丸め処理される。ここで、処理対象の符号化ブロックにおける動きベクトルの解像度が1/4画素精度の場合、MvShift=0である。同様に、動きベクトルの解像度が1画素精度の場合はMvShift=2であり、動きベクトルの解像度が4画素精度の場合はMvShift=4である。上式により、mvXのx,y成分それぞれが処理される。

【0320】

適応動きベクトル解像度は、サブブロック予測動きベクトルモードに対して用いることもできる。この場合、上記の通常予測動きベクトルモードに対して、解像度のみが異なる。すなわち、アフィン継承予測動きベクトル候補導出部361および461と、アフィン構築予測動きベクトル候補導出部362および462と、アフィン同一予測動きベクトル候補導出部363および463において、導出された候補の動きベクトルは解像度に応じて丸められる。解像度は1/16, 1/4, 1画素精度から選択でき、解像度を変更しない場合は1/16画素精度となる。丸め処理は、処理対象の符号化ブロックにおける動きベクトルの解像度に合わせてなされる。つまり、導出された候補の動きベクトルmvXは、上記の式により丸め処理される。ここで、処理対象の符号化ブロックにおける動きベクトルの解像度が1/4画素精度の場合、MvShift=0である。同様に、動きベクトルの解像度が1画素精度の場合はMvShift=2である。上記の式により、mvXのx,y成分それぞれが処理される。

(第2の実施の形態)

次に、第2の実施の形態に係る画像符号化装置および画像復号装置について説明する。第1の実施の形態に係る画像符号化装置および画像復号装置とは、構成が同じであるが、履歴マージ候補導出部345、445の処理手順が異なる。履歴マージ候補導出部345、445の処理手順である図42のフローチャートの代わりに図66のフローチャートとなっており、これらの違いについて説明する。

< 第2の実施の形態の履歴マージ候補導出処理 >

第2の実施の形態に係る画像符号化装置および画像復号装置の履歴マージ候補導出処理について、図66のフローチャートを用いて説明する。

【0321】

第2の実施の形態では、処理対象の符号化ブロックの左側に隣接するブロックA1から導出された空間マージ候補A1、及び右側に隣接するブロックB1から導出された空間マージ候補B1と履歴予測動きベクトル候補リストの後ろの要素(最近付化された要素)からオフセット値hMvpIdxOffsetで指定した個数の各要素のみを比較する点が、第1の実施の形態と異なる。第1の実施の形態に係る画像符号化装置および画像復号装置の履歴予測動きベクトル候補導出処理を示す図42のフローチャートとは、図42のステップS2306、S2307、S2309が図66ではステップS2326、S2327、S2329にそれぞれ変更されている点が異なり、それ以外は同一の処理である。第2の実施の形

態においても、まず、初期化処理を行う（図 6 6 のステップ S 2 3 0 1）。isPruned[i] の 0 から (numCurrMergeCand - 1) 番目のそれぞれの要素に FALSE の値を設定し、変数 numOrigMergeCand に現在のマージ候補リストに登録されている要素の数 numCurrMergeCand を設定する。

【 0 3 2 2 】

続いて、インデックス hMvpIdx の初期値を 1 に設定し、この初期値から NumHmvpCand まで、図 6 6 のステップ S 2 3 0 3 からステップ S 2 3 1 1 までの追加処理を繰り返す（図 6 6 のステップ S 2 3 0 2 ~ S 2 3 1 2）。現在のマージ候補リストに登録されている要素の数 numCurrMergeCand が (最大マージ候補数 MaxNumMergeCand - 1) 以下でなければ、マージ候補リストのすべての要素にマージ候補が追加されたので、本履歴マージ候補導出処理を終了する（図 6 6 のステップ S 2 3 0 3 : NO）。現在のマージ候補リストに登録されている要素の数 numCurrMergeCand が (最大マージ候補数 MaxNumMergeCand - 1) 以下の場合（図 6 6 のステップ S 2 3 0 3 : YES）、ステップ S 2 3 0 4 以降の処理を行う。

【 0 3 2 3 】

まず、sameMotion に FALSE（偽）の値を設定する（図 6 6 のステップ S 2 3 0 4）。続いて、インデックス hMvpIdx が所定のオフセット値 hMvpIdxOffset 以下の場合、すなわちマージ候補リストに含まれていない要素かどうかを確認する場合（図 4 2 のステップ S 2 3 0 5 の YES）、インデックス i の初期値を 0 に設定し、この初期値から、所定の上限値である 1 と numOrigMergeCand - 1 のいずれか小さい値まで図 6 6 のステップ S 2 3 2 7、S 2 3 0 8 の処理を行う（図 6 6 の S 2 3 2 6 ~ S 2 3 2 9）。ここで、第 2 の実施の形態では、処理対象の符号化ブロックの左側に隣接するブロック A 1 から導出された空間マージ候補 A 1、及び右側に隣接する空間マージ候補 B 1 と履歴予測動きベクトル候補リストの後ろの要素（最近付化された要素）からオフセット値 hMvpIdxOffset で指定した個数の各要素のみを比較する。所定の上限値が 1 であるのは、処理対象の符号化ブロックの左側に隣接するブロック A 1 から導出された空間マージ候補 A 1 または右側に隣接する空間マージ候補 B 1 がマージ候補リストの 0 から数えて 0 番目と 1 番目にしか格納される可能性がないからである。履歴予測動きベクトル候補リストの 0 から数えて (NumHmvpCand - hMvpIdx) 番目の要素 HmvpCandList[NumHmvpCand - hMvpIdx] と、空間マージ候補 A 1、および B 1 を比較する。（図 6 6 のステップ S 2 3 2 7）。マージ候補が持つすべての構成要素（インター予測モード、L 0 及び L 1 の参照インデックス、L 0 及び L 1 の動きベクトル）の値が同じかどうかを比較する。ここで、マージ候補が同じ値とは、マージ候補が持つすべての構成要素（インター予測モード、L 0 及び L 1 の参照インデックス、L 0 及び L 1 の動きベクトル）の値が同じであることを示す。したがって、このステップ S 2 3 2 7 の比較処理では、マージ候補リストの 0 から数えて i 番目の要素 mergeCandList[i] が左側に隣接するブロック A 1 から導出された空間マージ候補 A 1 または右側に隣接する空間マージ候補 B 1 で、かつ isPruned[i] が FALSE（偽）の際に、mergeCandList[i] と HmvpCandList[NumHmvpCand - hMvpIdx] が持つすべての構成要素（インター予測モード、L 0 及び L 1 の参照インデックス、L 0 及び L 1 の動きベクトル）の値が同じかどうかを比較する。同じ値の場合（図 3 9 のステップ S 2 3 2 7 : YES）、sameMotion および isPruned[i] 共に TRUE（真）を設定する（図 6 6 のステップ S 2 3 0 8）。なお、フラグ isPruned[i] はマージ候補リストの 0 から数えて i 番目の要素が履歴予測動きベクトル候補リストのいずれかの要素と同じ値であることを示すフラグである。同じ値でない場合（図 3 9 のステップ S 2 3 2 7 : NO）、ステップ S 2 3 0 8 の処理をスキップする。図 6 6 のステップ S 2 3 2 6 からステップ S 2 3 2 9 までの繰り返し処理が完了したら sameMotion が FALSE（偽）か否かを比較し（図 6 6 のステップ S 2 3 1 0）、sameMotion が FALSE（偽）の場合（図 6 6 のステップ S 2 3 1 0 : YES）、マージ候補リストの numCurrMergeCand 番目の mergeCandList[numCurrMergeCand] に履歴予測動きベクトル候補リストの 0 から数えて (NumHmvpCand - hMvpIdx) 番目の要素 HmvpCandList[NumHmvpCand - hMvpIdx] を追加し、numCurrMergeCand を 1 イン

クリメントする（図 6 6 のステップ S 2 3 1 1）。インデックス hMvpIdx を 1 インクリメントし（図 6 6 のステップ S 2 3 0 2）、図 6 6 のステップ S 2 3 0 2 ~ S 2 3 1 2 の繰り返し処理を行う。

【 0 3 2 4 】

履歴予測動きベクトル候補リストのすべての要素の確認が完了するか、マージ候補リストのすべての要素にマージ候補が追加されたら、本履歴マージ候補の導出処理を完了する。

【 0 3 2 5 】

なお、第 2 の実施の形態では、マージ候補リストに格納され空間マージ候補 A 1 および B 1 と、履歴予測動きベクトル候補リストの要素を比較するものとして説明したが、空間マージ候補 A 1 および B 1 をそれぞれマージ候補リスト以外のメモリに記憶し、マージ候補リスト以外のメモリに記憶された空間マージ候補 A 1 および B 1 と、履歴予測動きベクトル候補リストの要素を比較してもよい。

（第 3 の実施の形態）

次に、第 3 の実施の形態に係る画像符号化装置および画像復号装置について説明する。第 1 の実施の形態に係る画像符号化装置および画像復号装置とは、構成が同じであるが、符号化側の符号化情報格納メモリ 1 1 1 及び復号側の符号化情報格納メモリ 2 0 5 に備える履歴予測動きベクトル候補リスト初期化・更新処理手順における、同一要素確認処理手順が異なる。第 1 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順である図 3 9 のフローチャートの代わりに、第 3 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順では図 6 7 のフローチャートとなっており、これらの違いについて説明する。

< 第 3 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順 >

第 3 の実施の形態に係る画像符号化装置および画像復号装置の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順について、図 6 7 のフローチャートを用いて説明する。

【 0 3 2 6 】

第 3 の実施の形態では、履歴予測動きベクトル候補リスト更新処理において、履歴予測動きベクトル候補リスト HmvpCandList に要素が最大数追加されている場合、すなわち、現在の履歴予測動きベクトル候補の数 NumHmvpCand が履歴予測動きベクトル候補の最大数 MaxNumHmvpCand に達している場合、履歴予測動きベクトル候補リストに含まれる先頭の要素、すなわち 0 から数えて 0 番目の要素（履歴予測動きベクトル候補）を比較せず、1 番目以降の要素のみを比較する点が、第 1 の実施の形態と異なる。履歴予測動きベクトル候補リストに含まれる要素はインター予測モード、参照インデックス、動きベクトルで構成される。履歴予測動きベクトル候補リストに含まれる先頭の要素を比較しないことで、要素の比較回数は最大で（MaxNumHmvpCand - 1）回に制限され、要素の比較に伴う最大処理量を削減する。第 1 の実施の形態に係る画像符号化装置および画像復号装置の履歴予測動きベクトル候補導出処理を示す図 3 9 のフローチャートとは、図 3 9 のステップ S 2 1 2 2、S 2 1 2 5 が図 6 7 ではステップ S 2 1 3 2、2 1 3 5 にそれぞれ変更されている点が異なり、それ以外は同一の処理である。

【 0 3 2 7 】

第 3 の実施の形態においても、履歴予測動きベクトル候補の数 NumHmvpCand の値が 0 の場合（図 6 7 のステップ S 2 1 2 1：NO）、履歴予測動きベクトル候補リスト HmvpCandList は空で、同一候補は存在しないので図 6 7 ステップ S 2 1 3 2 ~ S 2 1 3 5 をスキップし、本同一要素確認処理手順を終了する。履歴予測動きベクトル候補の数 NumHmvpCand の値が 0 より大きい場合（図 6 7 のステップ S 2 1 2 1：YES）、履歴予測動きベクトルインデックス hMvpIdx が 0 または 1 から NumHmvpCand - 1 まで、ステップ S 2 1 2 3 の処理を繰り返す（図 6 7 のステップ S 2 1 3 2 ~ S 2 1 3 5）。まず、現在の履歴予測動きベクトル候補の数 NumHmvpCand が履歴予測動きベクトル候補の最大数 MaxNumHmvpCand 未満の場合、履歴予測動きベクトル候補リストに含まれる先頭の要素、す

なわち 0 から数えて 0 番目の要素（履歴予測動きベクトル候補）を比較するので、hMvpIdx を 0 とする。一方、現在の履歴予測動きベクトル候補の数 NumHmvpCand が履歴予測動きベクトル候補の最大数 MaxNumHmvpCand に達している場合、履歴予測動きベクトル候補リストに含まれる先頭の要素、すなわち 0 から数えて 0 番目の要素（履歴予測動きベクトル候補）を比較しないので、hMvpIdx を 1 とする（図 6 7 のステップ S 2 1 3 2）。続いて、履歴予測動きベクトル候補リストの 0 から数えて hMvpIdx 番目の要素 HmvpCandList[hMvpIdx] が登録対象のインター予測情報候補 hMvpCand と同一か否かを比較する（図 6 7 のステップ S 2 1 2 3）。同一の場合（図 6 7 のステップ S 2 1 2 3：YES）、同一候補が存在するか否かを示すフラグ identicalCandExist に TRUE（真）の値を設定し、削除対象インデックス removeIdx に hMvpIndex の値を設定し、本同一要素確認処理を終了する。同一でない場合（図 6 7 のステップ S 2 1 2 3：NO）、hMvpIdx を 1 インクリメントし、履歴予測動きベクトルインデックス hMvpIdx が NumHmvpCand - 1 以下であれば、ステップ S 2 1 2 3 以降の処理を行う（図 6 7 のステップ S 2 1 3 2 ~ S 2 1 3 5）。

（第 4 の実施の形態）

次に、第 4 の実施の形態に係る画像符号化装置および画像復号装置について説明する。第 4 の実施の形態に係る画像符号化装置および画像復号装置とは、構成が同じであるが、符号化側の符号化情報格納メモリ 1 1 1 及び復号側の符号化情報格納メモリ 2 0 5 に備える履歴予測動きベクトル候補リスト初期化・更新処理手順における、同一要素確認処理手順が異なる。第 1 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順である図 3 9 のフローチャートの代わりに、第 4 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順では図 6 8 のフローチャートとなっており、これらの違いについて説明する。

< 第 4 の実施形態の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順 >

第 4 の実施の形態に係る画像符号化装置および画像復号装置の履歴予測動きベクトル候補リスト初期化・更新処理手順における同一要素確認処理手順について、図 6 8 のフローチャートを用いて説明する。

【 0 3 2 8 】

第 4 の実施の形態では、履歴予測動きベクトル候補リスト更新処理において、履歴予測動きベクトル候補リストの最後の要素から降順に要素を比較する点が、第 1 の実施の形態、及び第 3 の実施の形態と異なる。さらに、履歴予測動きベクトル候補リスト更新処理において、履歴予測動きベクトル候補リスト HmvpCandList に要素が最大数追加されている場合、履歴予測動きベクトル候補リストに含まれる先頭の要素、すなわち 0 から数えて 0 番目の要素を比較せず、1 番目以降の要素のみを比較する点が、第 1 の実施の形態と異なる。履歴予測動きベクトル候補リストに含まれる要素はインター予測モード、参照インデックス、動きベクトルで構成される。履歴予測動きベクトル候補リストに含まれる先頭の要素を比較しないことで、要素の比較回数は最大で（MaxNumHmvpCand - 1）回に制限され、要素の比較に伴う最大処理量を削減する。

【 0 3 2 9 】

第 4 の実施の形態においても、履歴予測動きベクトル候補の数 NumHmvpCand の値が 0 の場合（図 6 8 のステップ S 2 1 5 1：NO）、履歴予測動きベクトル候補リスト HmvpCandList は空で、同一候補は存在しないので図 6 8 ステップ S 2 1 5 2 ~ S 2 1 5 5 をスキップし、本同一要素確認処理手順を終了する。履歴予測動きベクトル候補の数 NumHmvpCand の値が 0 より大きい場合（図 6 8 のステップ S 2 1 5 2：YES）、インデックス i が 1 から、履歴予測動きベクトル候補の最大数 MaxNumHmvpCand - 1 と履歴予測動きベクトル候補の数 NumHmvpCand のいずれか小さい値まで、ステップ S 2 1 5 3 の処理を繰り返す（図 6 8 のステップ S 2 1 5 2 ~ S 2 1 5 5）。まず、履歴予測動きベクトル候補リストの 0 から数えて NumHmvpCand - i 番目の要素 HmvpCandList[NumHmvpCand - i] が登録対象のインター予測情報候補 hMvpCand と同一か否かを比較する（図 6 8 の

ステップS 2 1 5 3)。同一の場合（図68のステップS 2 1 5 3：YES）、同一候補が存在するか否かを示すフラグidenticalCandExistにTRUE（真）の値を設定し、削除対象インデックスremoveldxにhMVpIndexの値を設定し、本同一要素確認処理を終了する。同一でない場合（図68のステップS 2 1 5 3：NO）、iを1インクリメントし、iが履歴予測動きベクトル候補の最大数MaxNumHmvpCand -1と履歴予測動きベクトル候補の数NumHmvpCandのいずれか小さい値以下であれば、ステップS 2 1 5 3以降の処理を行う（図68のステップS 2 1 5 2～S 2 1 5 5）。なお、インデックスiが初期値である1の時、履歴予測動きベクトル候補リストの0から数えてNumHmvpCand-i番目の要素HmvpCandList[NumHmvpCand - i]は履歴予測動きベクトル候補リストに登録されている最後の要素を示し、インデックスiが1ずつインクリメントされるにつれて履歴予測動きベクトル候補リストの要素を降順に示す。インデックスiは最大で(MaxNumHmvpCand -1)とすることで、履歴予測動きベクトル候補の先頭の要素HmvpCandList[0]と比較することはない。

10

【0330】

以上に述べた全ての実施の形態は、複数を組み合わせても良い。

【0331】

以上に述べた全ての実施の形態において、画像符号化装置が出力する符号化ビットストリームは、実施の形態で用いられた符号化方法に応じて復号することができるように特定のデータフォーマットを有している。符号化ビットストリームは、HDD、SSD、フラッシュメモリ、光ディスク等のコンピュータ等で読み解き可能な記録媒体に記録して提供しても良いし、有線あるいは無線のネットワークを通してサーバから提供しても良い。従って、この画像符号化装置に対応する画像復号装置は、提供手段によらず、この特定のデータフォーマットの符号化ビットストリームを復号することができる。

20

【0332】

画像符号化装置と画像復号装置の間で符号化ビットストリームをやりとりするために、有線または無線のネットワークが用いられる場合、通信路の伝送形態に適したデータ形式に符号化ビットストリームを変換して伝送してもよい。その場合、画像符号化装置が出力する符号化ビットストリームを通信路の伝送形態に適したデータ形式の符号化データに変換してネットワークに送信する送信装置と、ネットワークから符号化データを受信して符号化ビットストリームに復元して画像復号装置に供給する受信装置とが設けられる。送信装置は、画像符号化装置が出力する符号化ビットストリームをバッファするメモリと、符号化ビットストリームをパケット化するパケット処理部と、ネットワークを介してパケット化された符号化データを送信する送信部とを含む。受信装置は、ネットワークを介してパケット化された符号化データを受信する受信部と、受信された符号化データをバッファするメモリと、符号化データをパケット処理して符号化ビットストリームを生成し、画像復号装置に提供するパケット処理部とを含む。

30

【0333】

画像符号化装置と画像復号装置の間で符号化ビットストリームをやりとりするために、有線または無線のネットワークが用いられる場合、送信装置、受信装置に加え、さらに、送信装置が送信する符号化データを受信し、受信装置に供給する中継装置が設けられても良い。中継装置は、送信装置が送信するパケット化された符号化データを受信する受信部と、受信された符号化データをバッファするメモリと、パケット化された符号化データとネットワークに送信する送信部とを含む。さらに、中継装置は、パケット化された符号化データをパケット処理して符号化ビットストリームを生成する受信パケット処理部と、符号化ビットストリームを蓄積する記録媒体と、符号化ビットストリームをパケット化する送信パケット処理部を含んでも良い。

40

【0334】

また、画像復号装置で復号された画像を表示する表示部を構成に追加することで、表示装置としても良い。その場合、表示部は、復号画像信号重畳部207により生成され、復号画像メモリ208に格納された復号画像信号を読み出して画面に表示する。

50

【 0 3 3 5 】

また、撮像部を構成に追加し、撮像した画像を画像符号化装置に入力することで、撮像装置としても良い。その場合、撮像部は、撮像した画像信号をブロック分割部 1 0 1 に入力する。

【 0 3 3 6 】

図 6 9 に、本願の符号化復号装置のハードウェア構成の一例を示す。符号化復号装置は、本発明の実施の形態に係る画像符号化装置、および画像復号装置の構成を包含する。係る符号化復号装置 9 0 0 0 は、CPU 9 0 0 1、コーデック IC 9 0 0 2、I/O インターフェース 9 0 0 3、メモリ 9 0 0 4、光学ディスクドライブ 9 0 0 5、ネットワークインターフェース 9 0 0 6、ビデオインターフェース 9 0 0 9 を有し、各部はバス 9 0 1 0 により接続される。

10

【 0 3 3 7 】

画像符号化部 9 0 0 7 と画像復号部 9 0 0 8 は、典型的にはコーデック IC 9 0 0 2 として実装される。本発明の実施の形態に係る画像符号化装置の画像符号化処理は、画像符号化部 9 0 0 7 により実行され、本発明の実施の形態に係る画像復号装置における画像復号処理は、画像符号化部 9 0 0 7 により実行される。I/O インターフェース 9 0 0 3 は、例えば USB インターフェースにより実現され、外部のキーボード 9 1 0 4、マウス 9 1 0 5 等と接続する。CPU 9 0 0 1 は、I/O インターフェース 9 0 0 3 を介して入力したユーザー操作に基づき、ユーザーの所望する動作を実行するように符号化復号装置 9 0 0 0 を制御する。キーボード 9 1 0 4、マウス 9 1 0 5 等によるユーザーの操作としては、符号化、復号のどちらの機能を実行するかを選択、符号化品質の設定、符号化ストリームの入出力先、画像の入出力先等がある。

20

【 0 3 3 8 】

ユーザーがディスク記録媒体 9 1 0 0 に記録された画像を再生する操作を所望する場合、光学ディスクドライブ 9 0 0 5 は、挿入されたディスク記録媒体 9 1 0 0 から符号化ビットストリームを読み出し、読み出した符号化ストリームを、バス 9 0 1 0 を介してコーデック IC 9 0 0 2 の画像復号部 9 0 0 8 に送る。画像復号部 9 0 0 8 は入力した符号化ビットストリームに対して本発明の実施の形態に係る画像復号装置における画像復号処理を実行し、復号画像を、ビデオインターフェース 9 0 0 9 を介して外部のモニター 9 1 0 3 へ送る。また、符号化復号装置 9 0 0 0 は、ネットワークインターフェース 9 0 0 6 を有し、ネットワーク 9 1 0 1 を介して、外部の配信サーバ 9 1 0 6 や、携帯端末 9 1 0 7 と接続可能である。ユーザーがディスク記録媒体 9 1 0 0 に記録された画像に変えて、配信サーバ 9 1 0 6 や携帯端末 9 1 0 7 に記録された画像を再生することを所望する場合は、ネットワークインターフェース 9 0 0 6 は、入力されたディスク記録媒体 9 1 0 0 から符号化ビットストリームを読み出すことに変えて、ネットワーク 9 1 0 1 より符号化ストリームを取得する。また、ユーザーがメモリ 9 0 0 4 に記録された画像を再生することを所望する場合は、メモリ 9 0 0 4 に記録された符号化ストリームに対して、本発明の実施の形態に係る画像復号装置における画像復号処理を実行する。

30

【 0 3 3 9 】

ユーザーが外部のカメラ 9 1 0 2 で撮像した画像を符号化しメモリ 9 0 0 4 に記録する操作を所望する場合、ビデオインターフェース 9 0 0 9 は、カメラ 9 1 0 2 から画像を入力し、バス 9 0 1 0 を介し、コーデック IC 9 0 0 2 の画像符号化部 9 0 0 7 に送る。画像符号化部 9 0 0 7 は、ビデオインターフェース 9 0 0 9 を介して入力した画像に対して本発明の実施の形態に係る画像符号化装置における画像符号化処理を実行し、符号化ビットストリームを作成する。そして符号化ビットストリームを、バス 9 0 1 0 を介し、メモリ 9 0 0 4 へ送る。ユーザーがメモリ 9 0 0 4 に変えて、ディスク記録媒体 9 1 0 0 に符号化ストリームを記録することを所望する場合は、光学ディスクドライブ 9 0 0 5 は、挿入されたディスク記録媒体 9 1 0 0 に対し符号化ストリームの書き出しを行う。

40

【 0 3 4 0 】

画像符号化装置を有し画像復号装置を有さないハードウェア構成や、画像復号装置を有

50

し画像符号化装置を有さないハードウェア構成を実現することも可能である。そのようなハードウェア構成は、例えばコーデックIC9002が、画像符号化部9007、または画像復号部9008にそれぞれ置き換わることにより実現される。

【0341】

以上の符号化及び復号に関する処理は、ハードウェアを用いた伝送、蓄積、受信装置として実現しても良いのは勿論のこと、ROM(リード・オンリー・メモリ)やフラッシュメモリ等に記憶されているファームウェアや、コンピュータ等のソフトウェアによって実現しても良い。そのファームウェアプログラム、ソフトウェアプログラムをコンピュータ等で読み取り可能な記録媒体に記録して提供しても良いし、有線あるいは無線のネットワークを通してサーバから提供しても良いし、地上波あるいは衛星デジタル放送のデータ

10

【0342】

以上、本発明を実施の形態をもとに説明した。実施の形態は例示であり、それらの各構成要素や各処理プロセスの組み合わせにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【符号の説明】

【0343】

100 画像符号化装置、 101 ブロック分割部、 102 インター予測部、 103 イントラ予測部、 104 復号画像メモリ、 105 予測方法決定部、 106 残差信号生成部、 107 直交変換・量子化部、 108 ビット列符号化部、 109 逆量子化・逆直交変換部、 110 復号画像信号重畳部、 111 符号化情報格納メモリ、 200 画像復号装置、 201 ビット列復号部、 202 ブロック分割部、 203 インター予測部、 204 イントラ予測部、 205 符号化情報格納メモリ、 206 逆量子化・逆直交変換部、 207 復号画像信号重畳部、 208 復号画像メモリ。

20

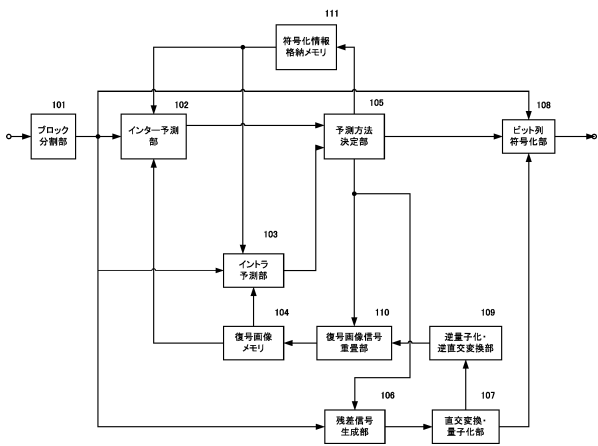
30

40

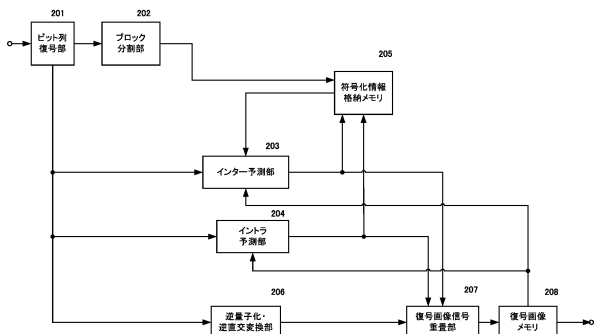
50

【図面】

【図 1】

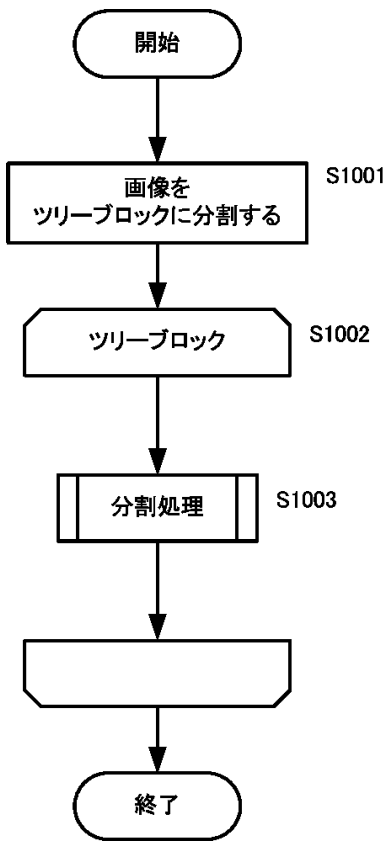


【図 2】

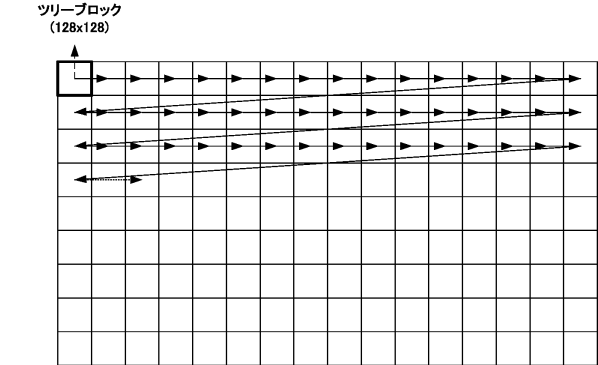


10

【図 3】



【図 4】



20

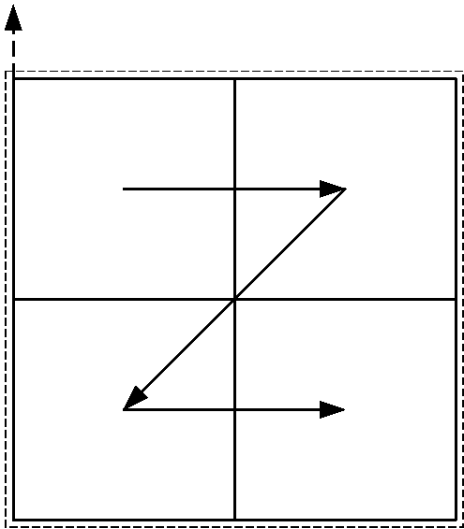
30

40

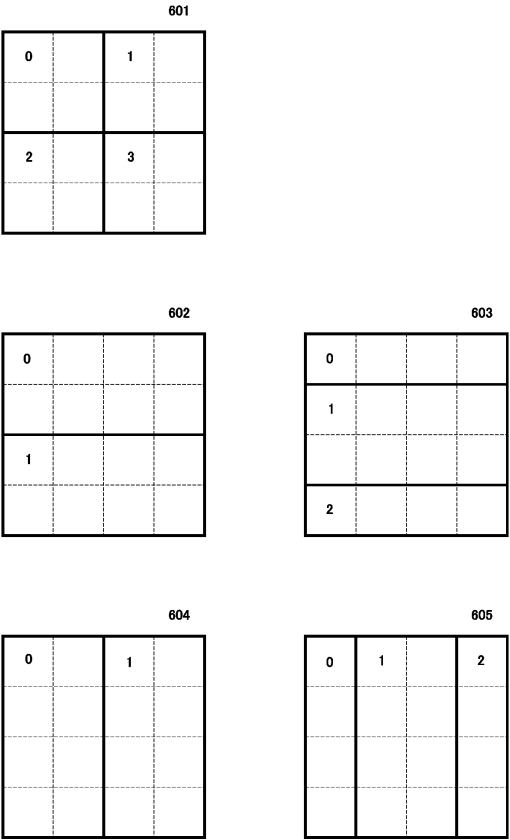
50

【図 5】

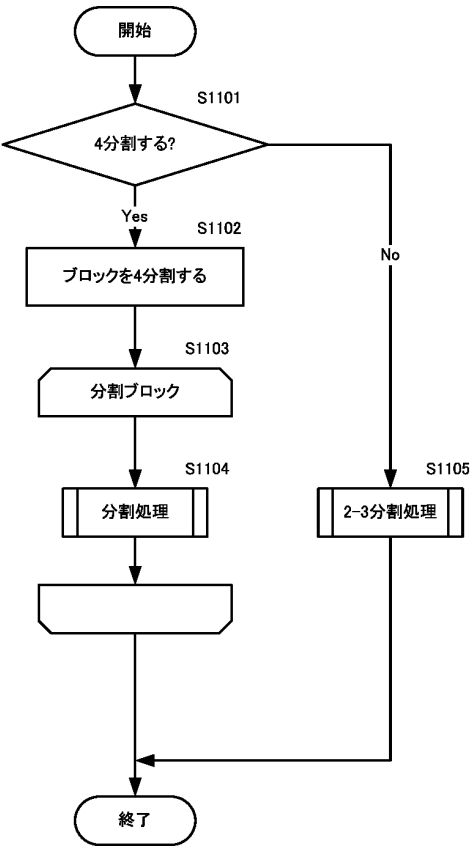
ツリーブロック
(128x128)



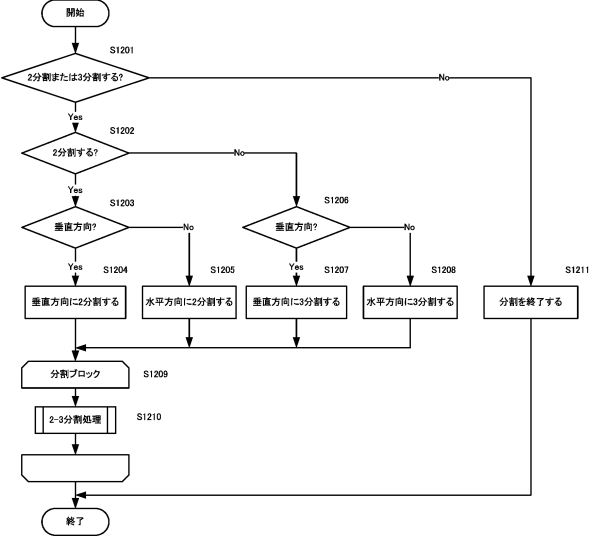
【図 6】



【図 7】



【図 8】



10

20

30

40

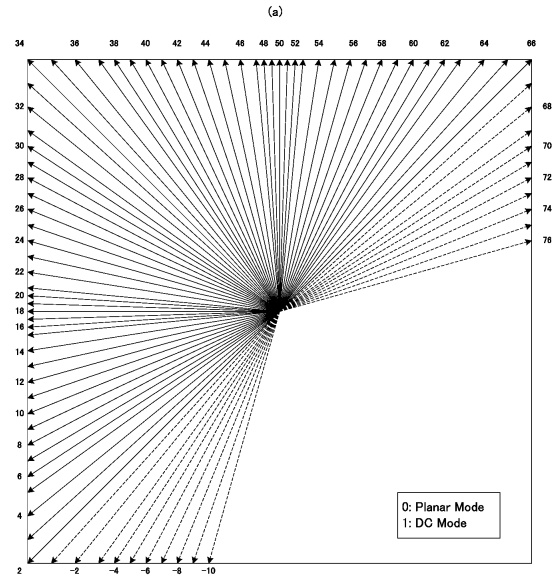
50

【図 9】

```
coding_quadtree() {
  qt_split
  if(qt_split) {
    coding_quadtree(0)
    coding_quadtree(1)
    coding_quadtree(2)
    coding_quadtree(3)
  }
  else {
    multi_type_tree()
  }
}
```

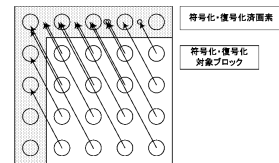
```
multi_type_tree() {
  mtt_split
  if(mtt_split) {
    mtt_split_vertical
    mtt_split_binary
    if(mtt_split_binary) {
      multi_type_tree(0, mtt_split_vertical)
      multi_type_tree(1, mtt_split_vertical)
    } else {
      multi_type_tree(0, mtt_split_vertical)
      multi_type_tree(1, mtt_split_vertical)
      multi_type_tree(2, mtt_split_vertical)
    }
  } else {
    // end split
  }
}
```

【図 10】



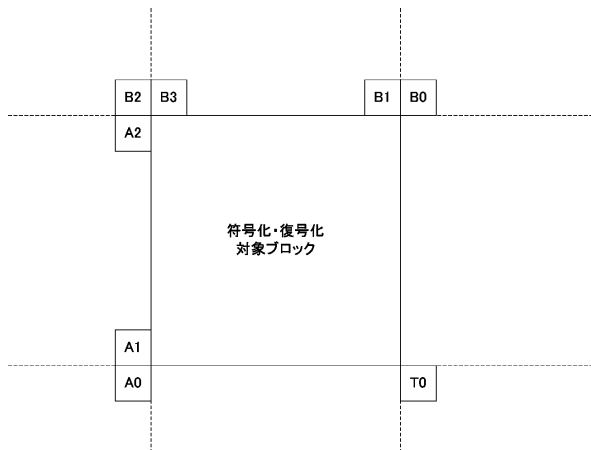
10

(b)



20

【図 11】



【図 12】

```
coding_unit() {
  pred_mode_flag
  if(MODE_INTRA) {
    intra_pred_mode
  }
  else { // MODE_INTER
    merge_flag
    if(merge_flag) {
      merge_affine_flag
      if(merge_affine_flag==0) {
        umve_flag
      }
    } else {
      inter_affine_flag
      if(inter_affine_flag) {
        cu_affine_type_flag
      }
    }
  }
}
```

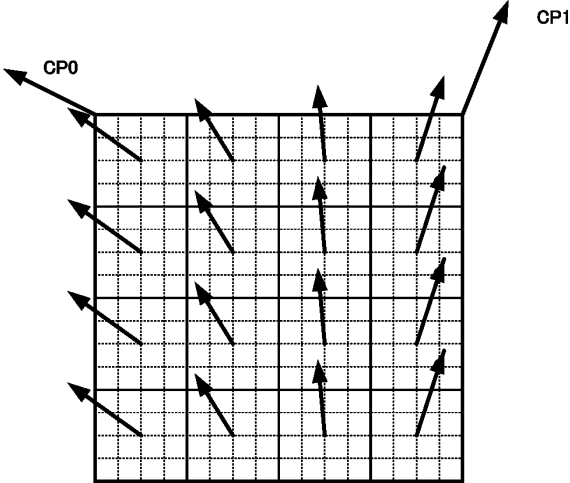
30

40

【図 1 3】

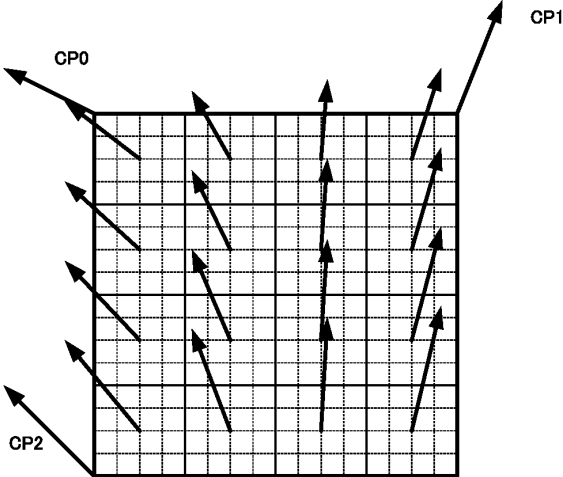
merge_flag	merge_affine_flag	inter_affine_flag	Selected Mode
1	0	N/A	Merge Mode
1	1	N/A	Affine Merge Mode
0	N/A	0	Inter Pred Mode
0	N/A	1	Inter Affine Mode

【図 1 4】

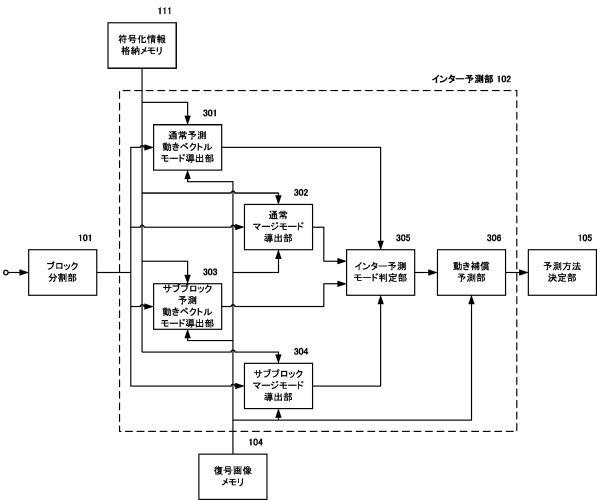


10

【図 1 5】



【図 1 6】



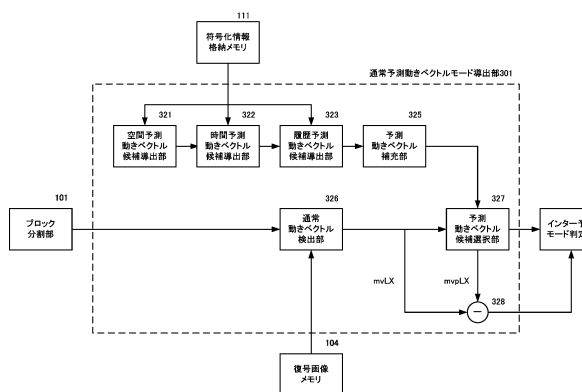
20

30

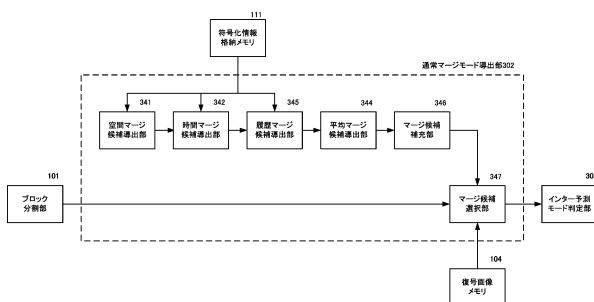
40

50

【图 17】

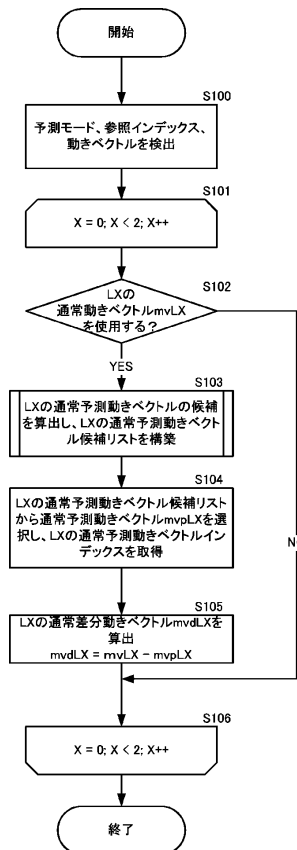


【 図 1 8 】

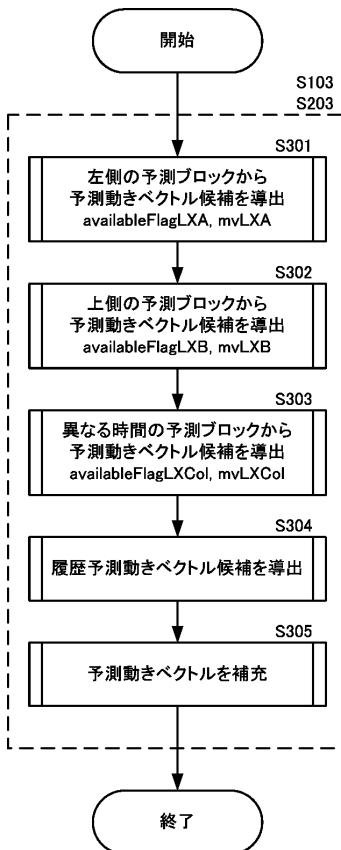


10

【 図 1 9 】



【 図 2 0 】



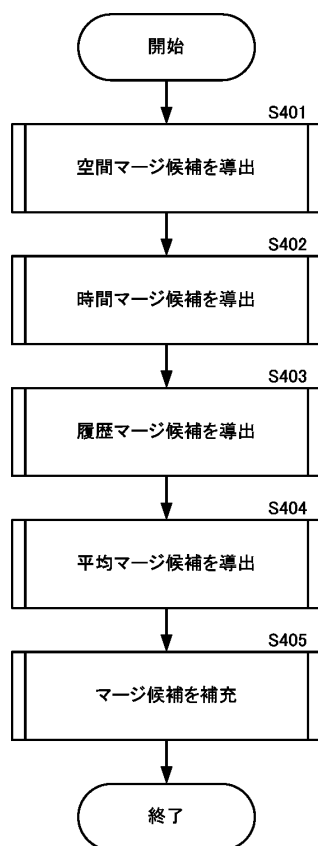
20

30

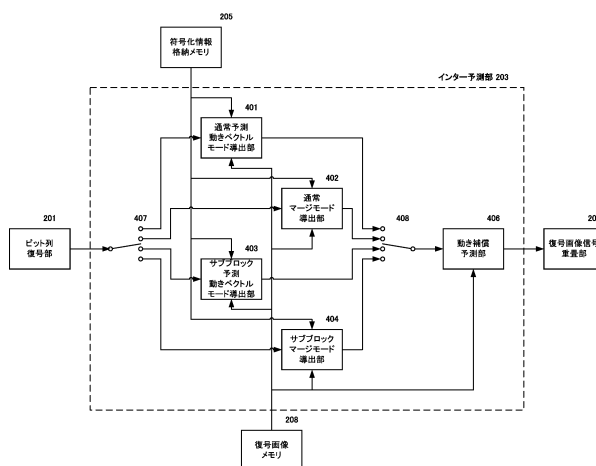
40

50

【 図 2 1 】



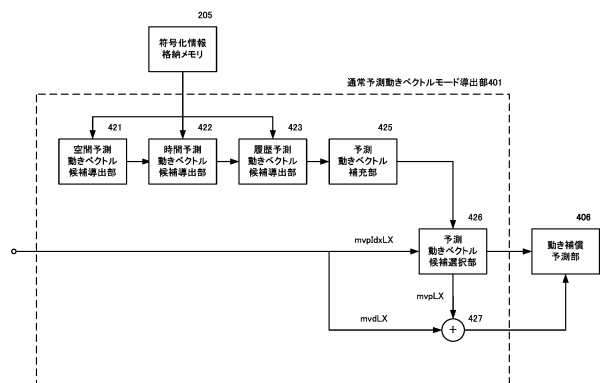
【 ㄨ 2 2 】



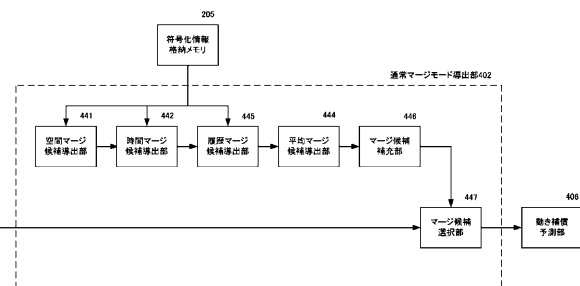
10

20

【 図 2 3 】



【 図 2 4 】

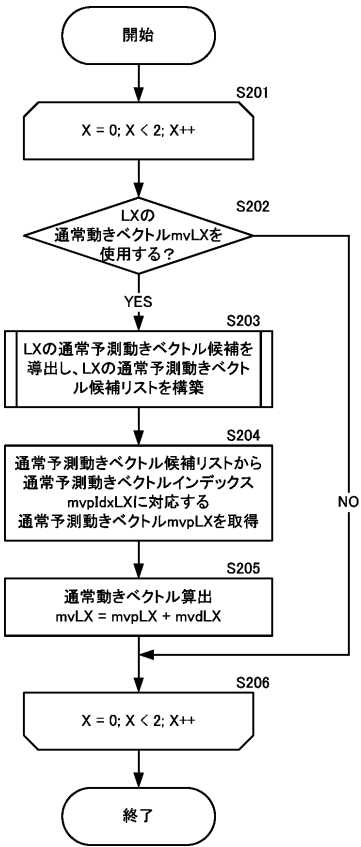


30

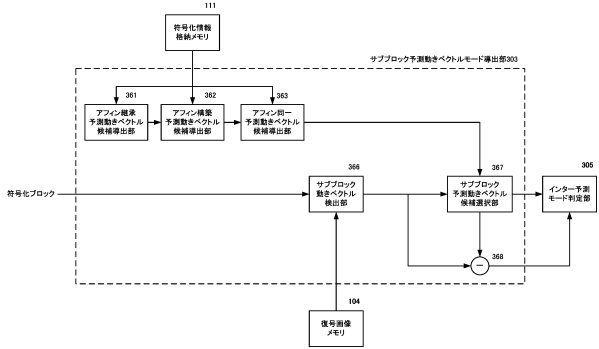
40

50

【図 25】



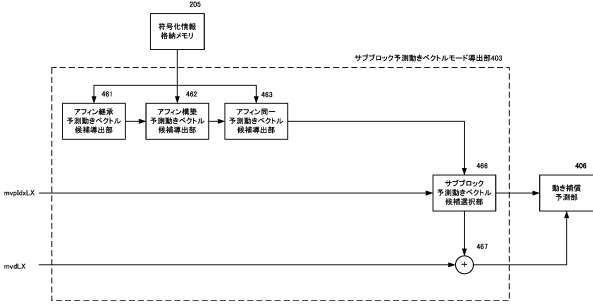
【図 26】



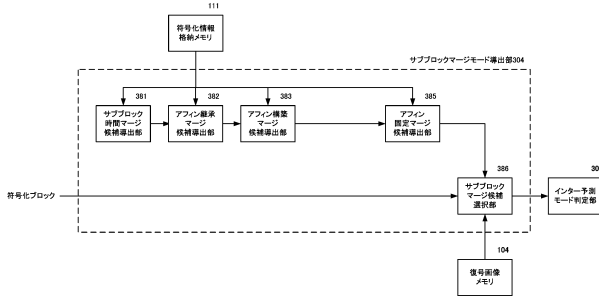
10

20

【図 27】



【図 28】

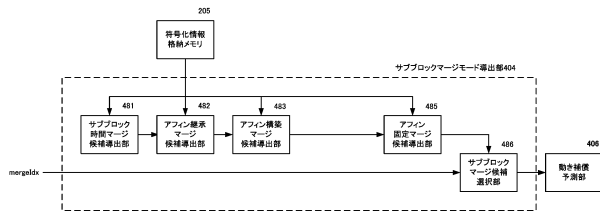


30

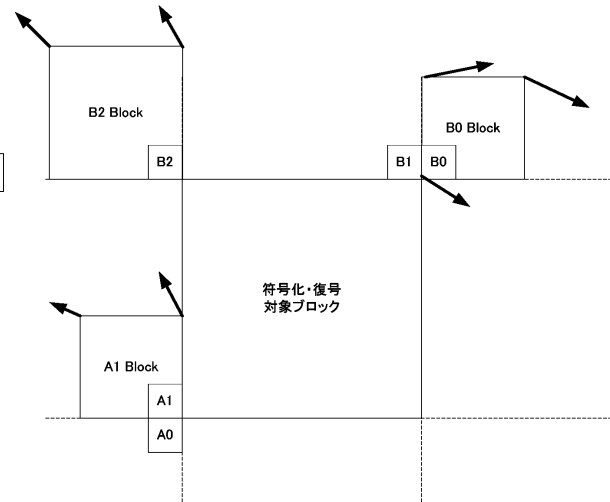
40

50

【図 29】

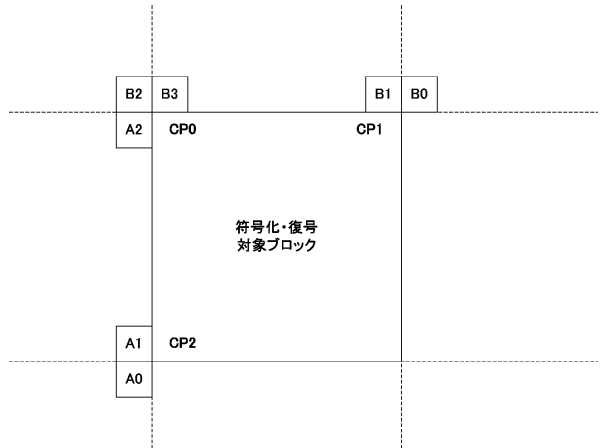


【図 30】

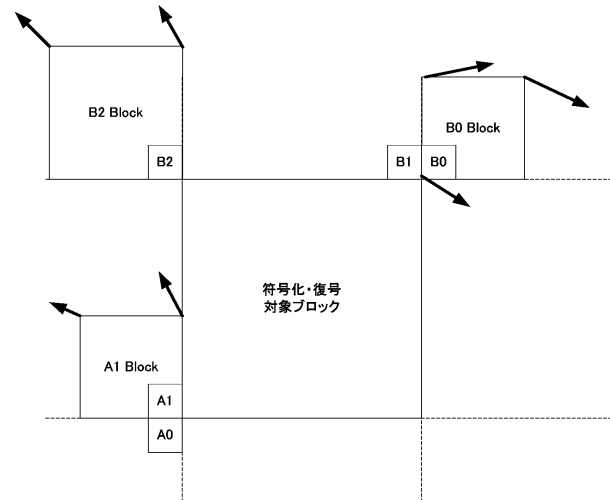


10

【図 31】



【図 32】



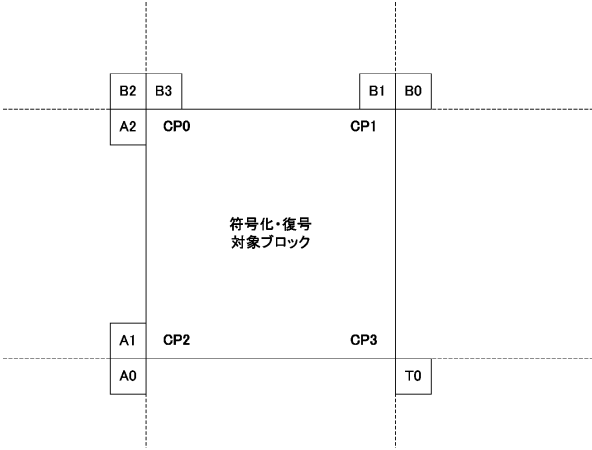
20

30

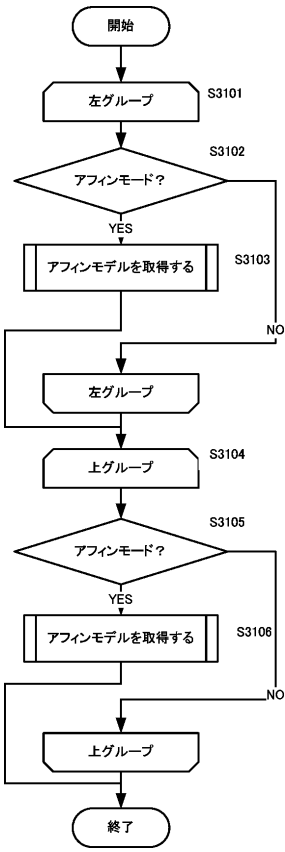
40

50

【図 3 3】



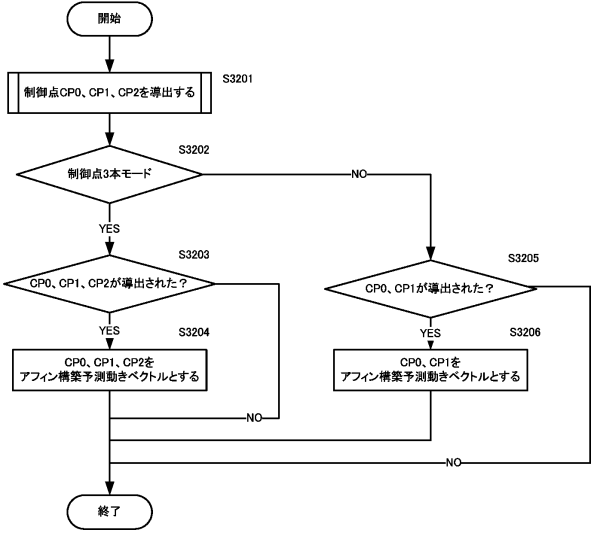
【図 3 4】



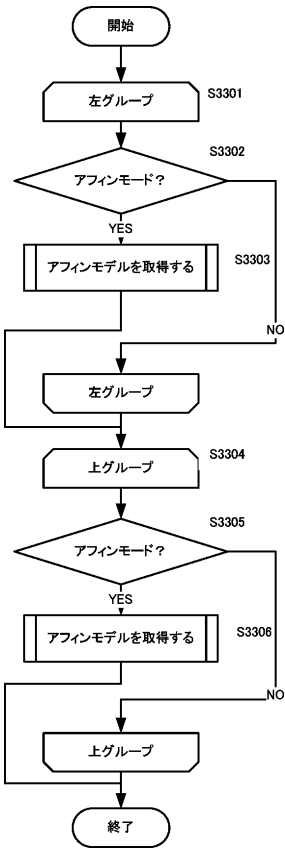
10

20

【図 3 5】



【図 3 6】

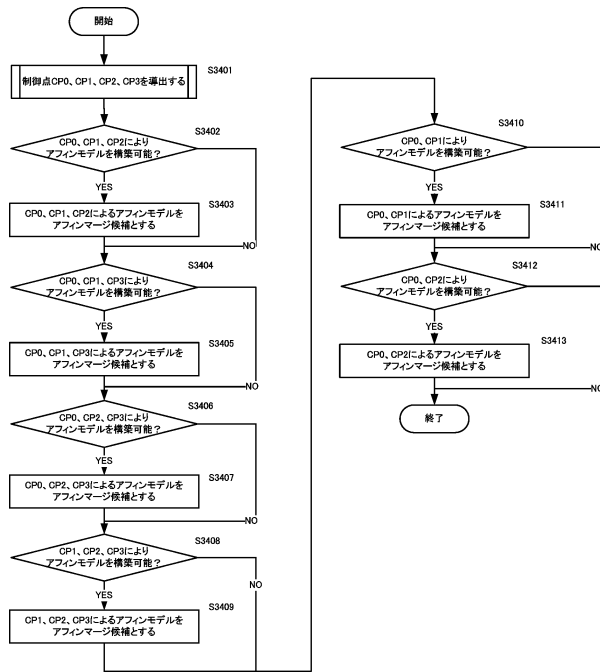


30

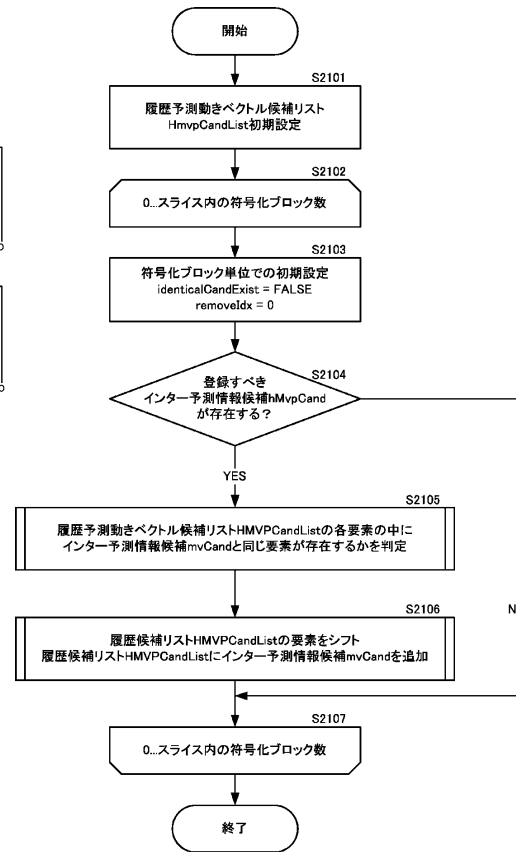
40

50

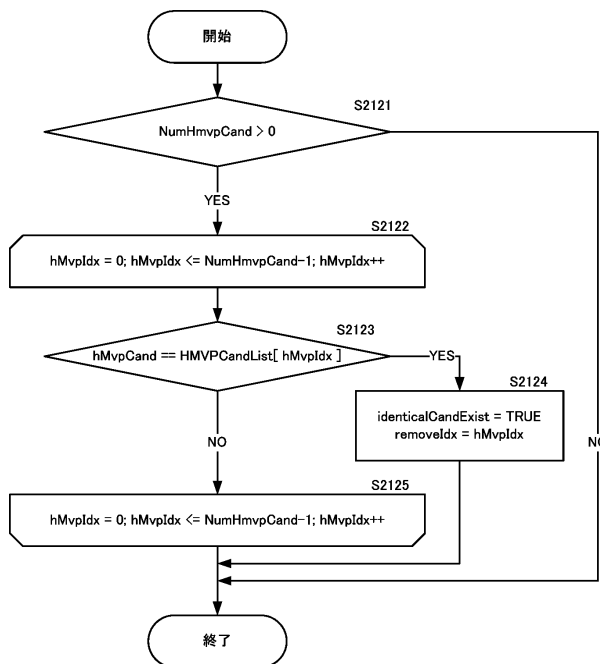
【図 37】



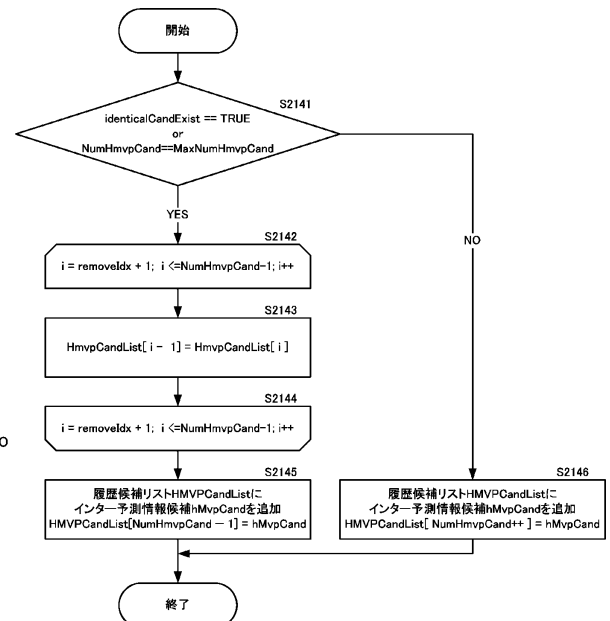
【図 38】



【図 39】



【図 40】



10

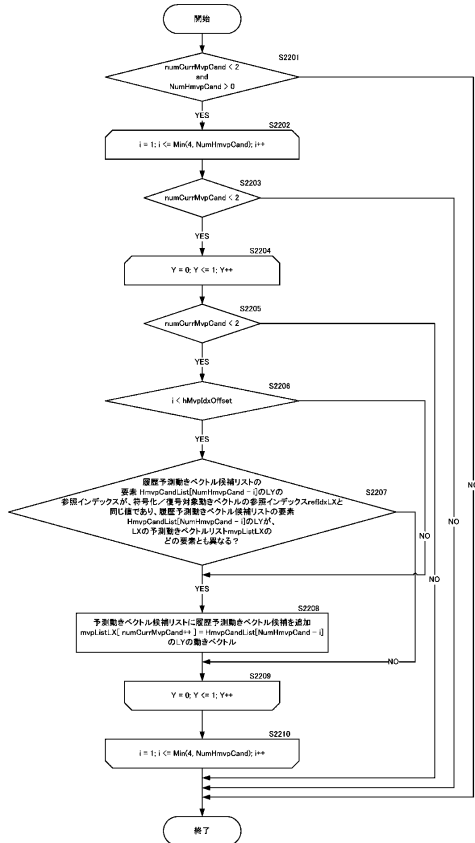
20

30

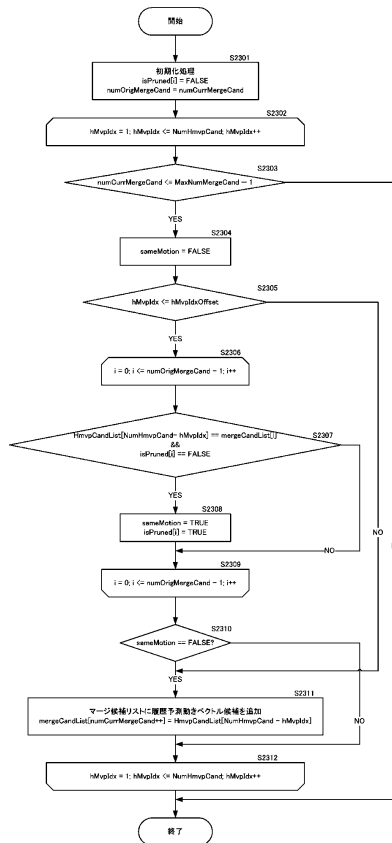
40

50

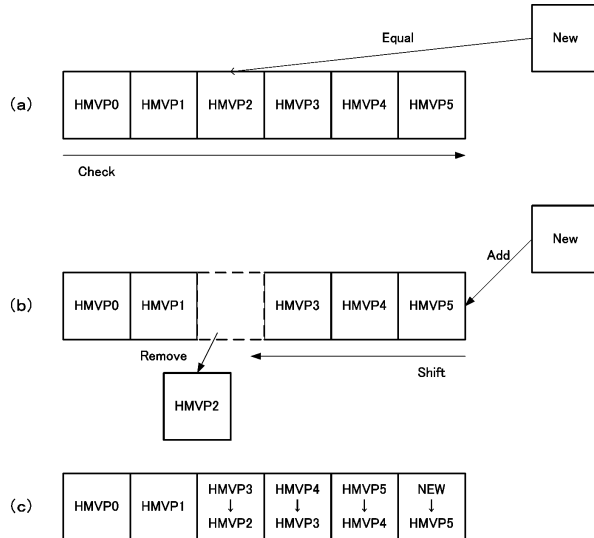
【図 4 1】



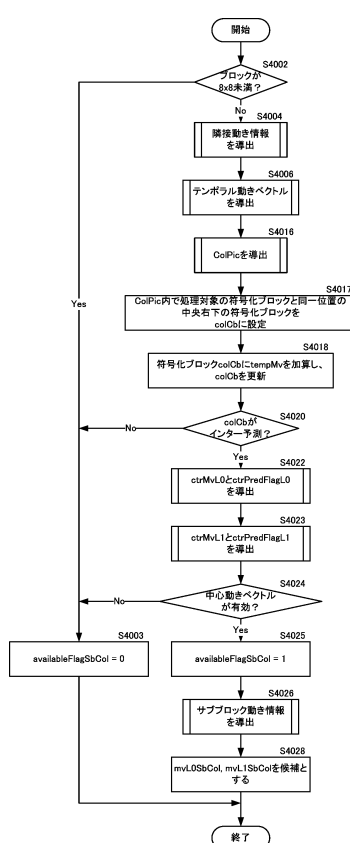
【図 4 2】



【図 4 3】



【図 4 4】



10

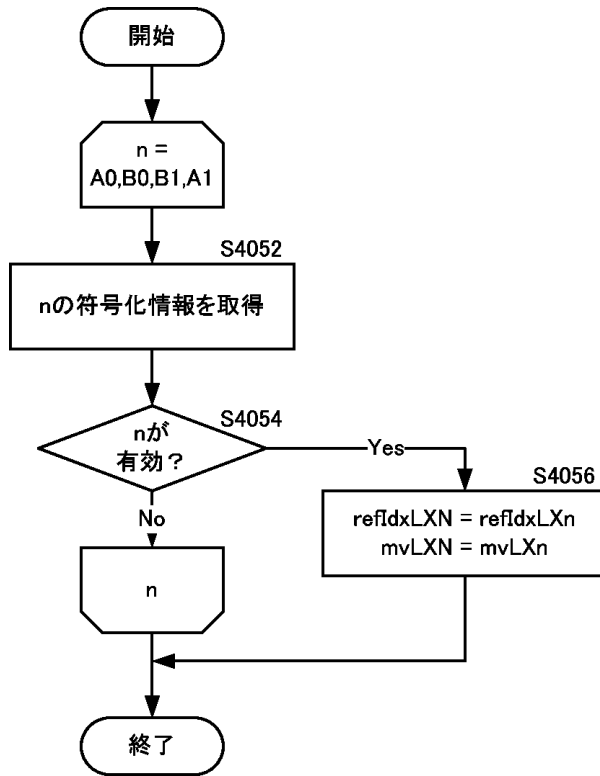
20

30

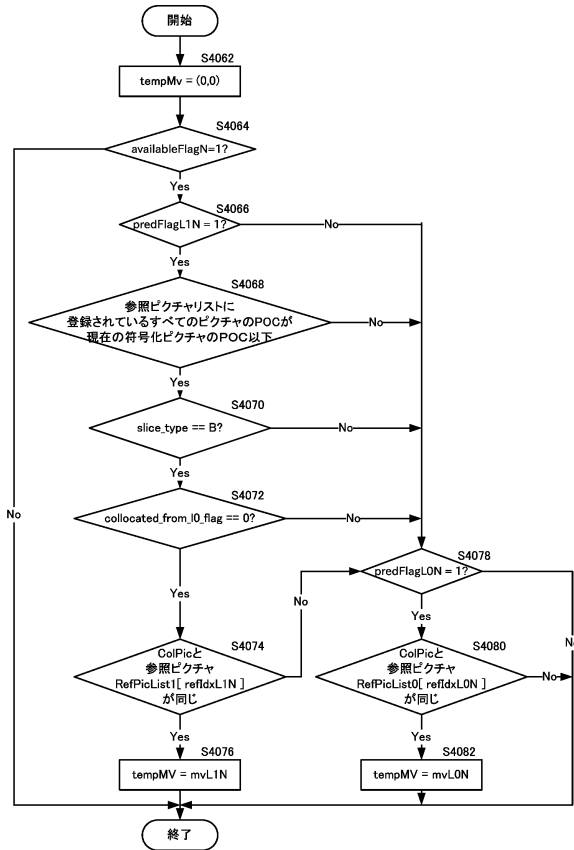
40

50

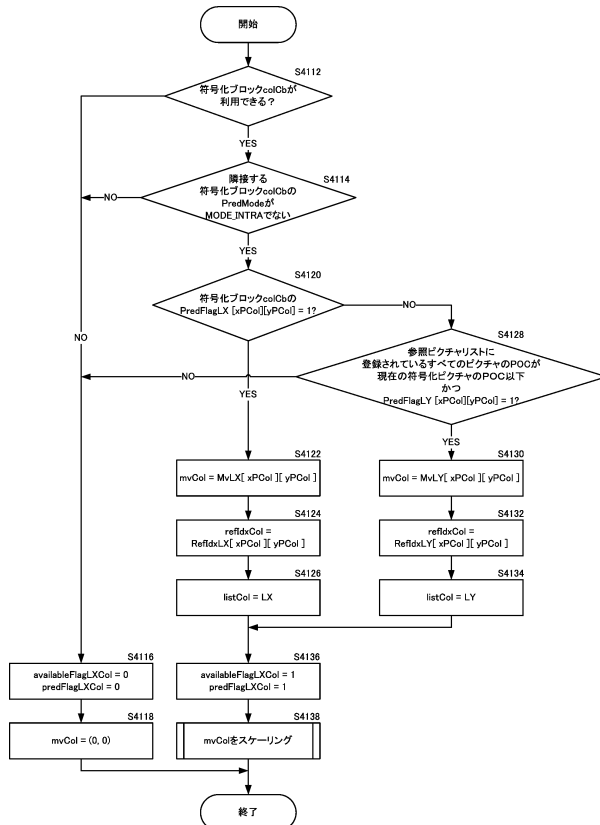
【図 4 5】



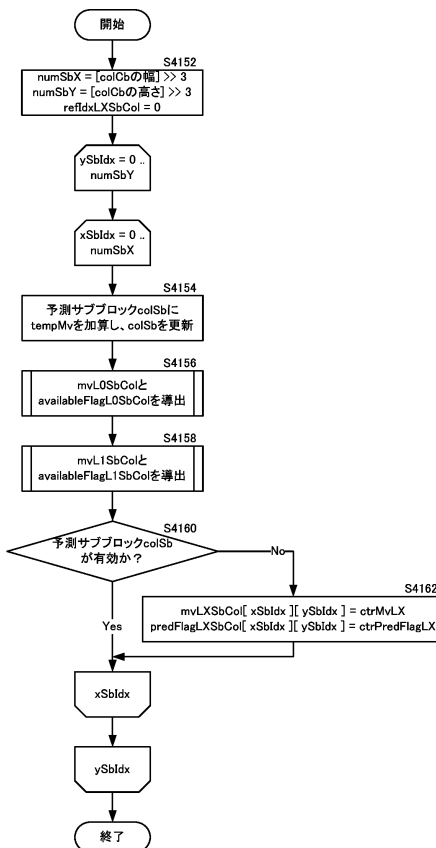
【図 4 6】



【図 4 7】



【図 4 8】



10

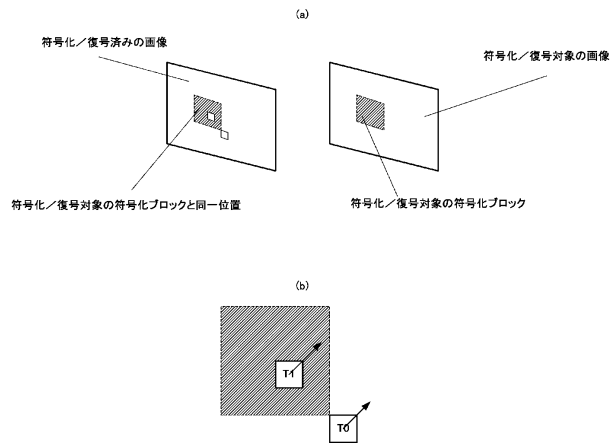
20

30

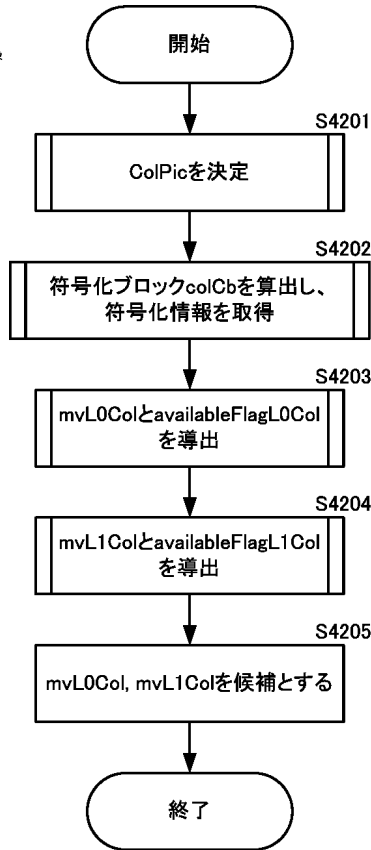
40

50

【図 49】



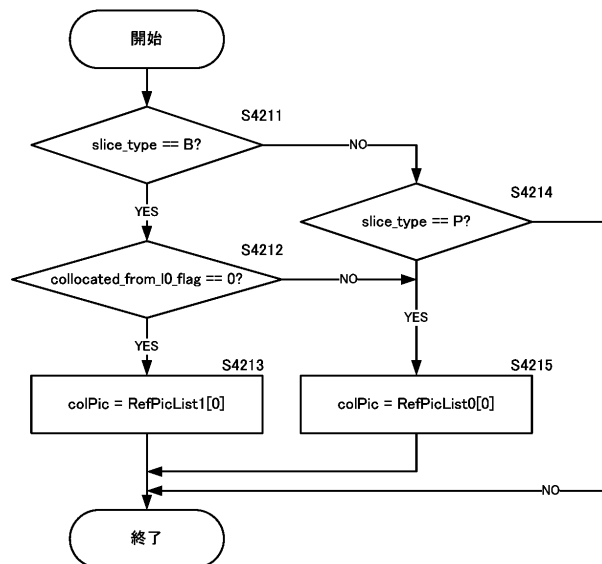
【図 50】



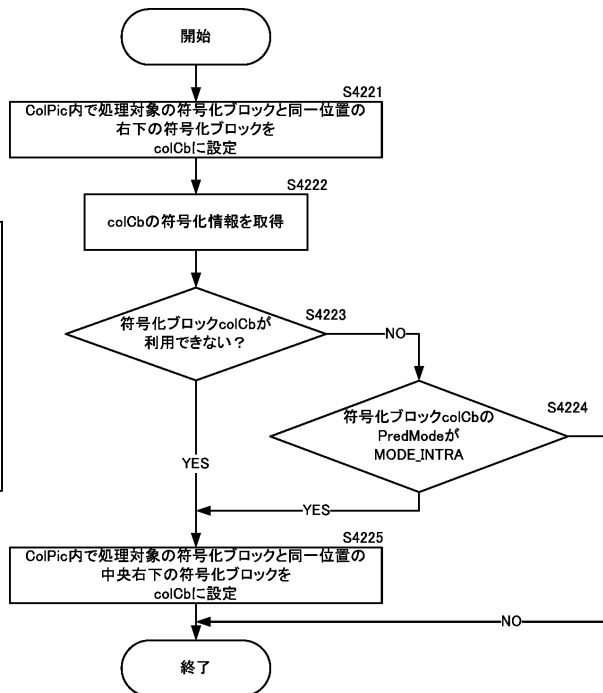
10

20

【図 51】



【図 52】

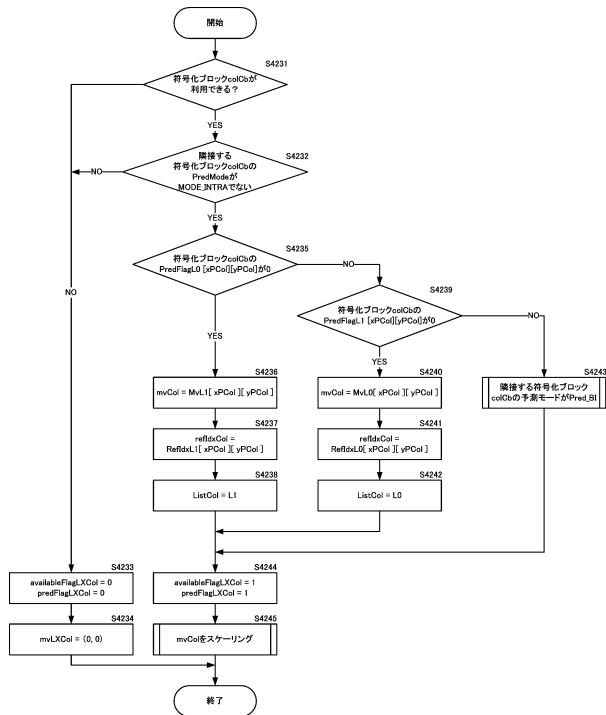


30

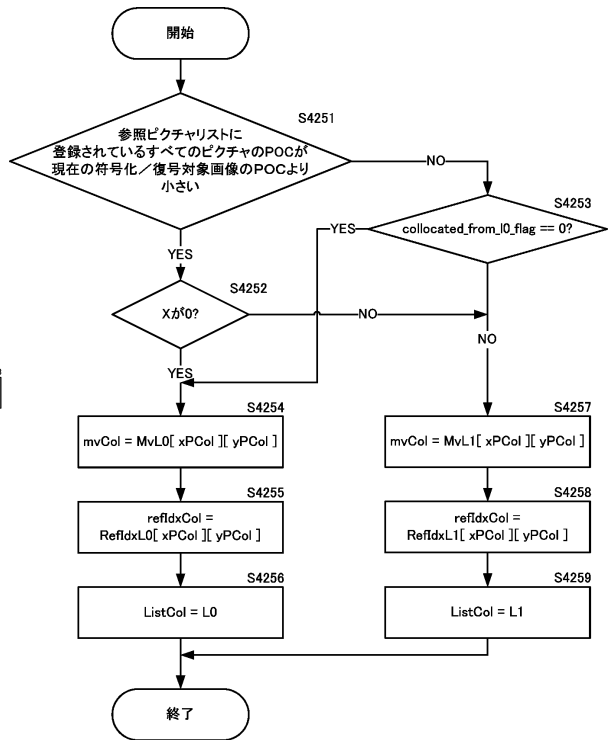
40

50

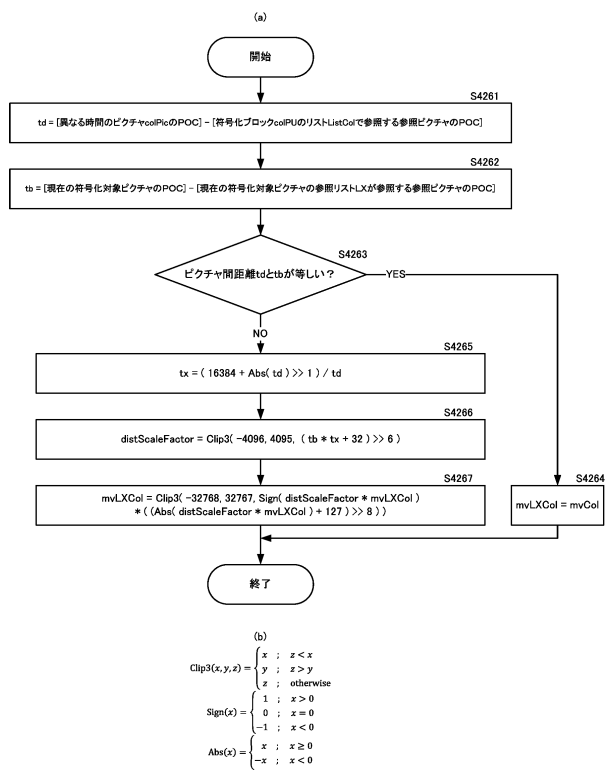
【図 5 3】



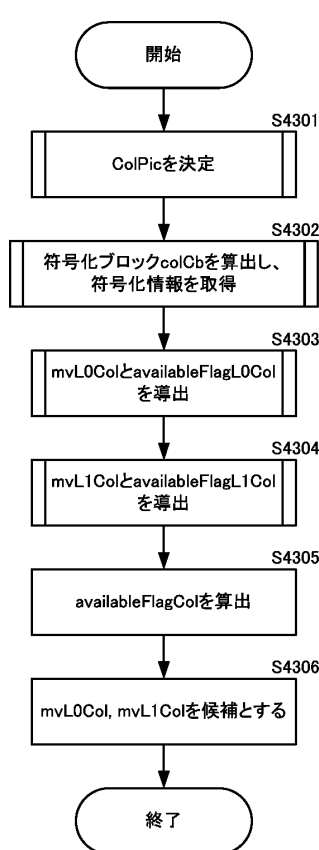
【図 5 4】



【図 5 5】



【図 5 6】



10

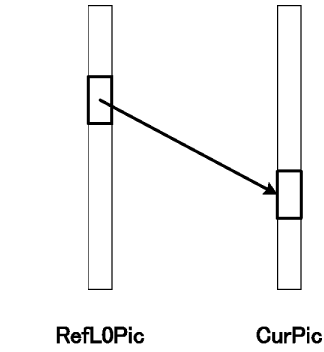
20

30

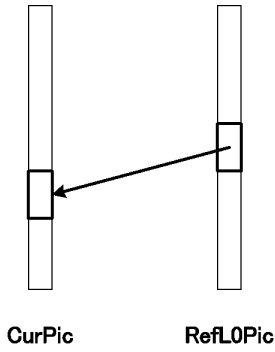
40

50

【図 5 7】

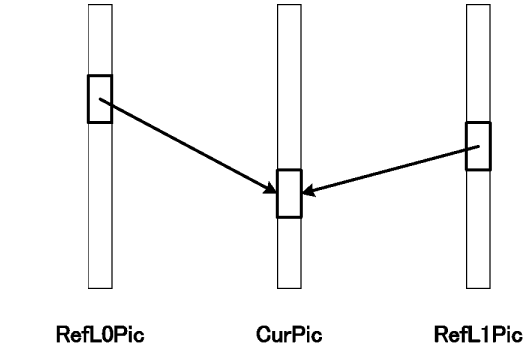


【図 5 8】

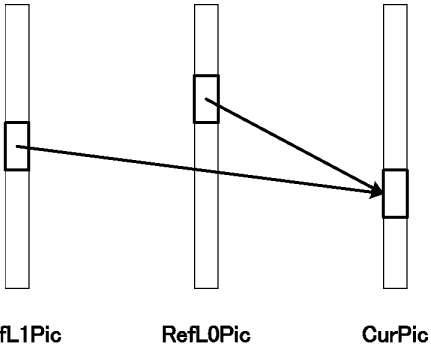


10

【図 5 9】



【図 6 0】



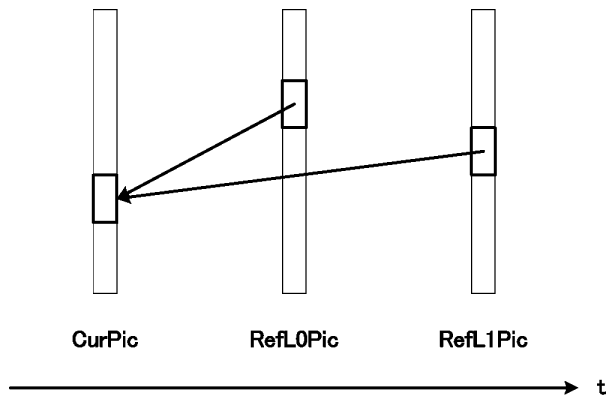
20

30

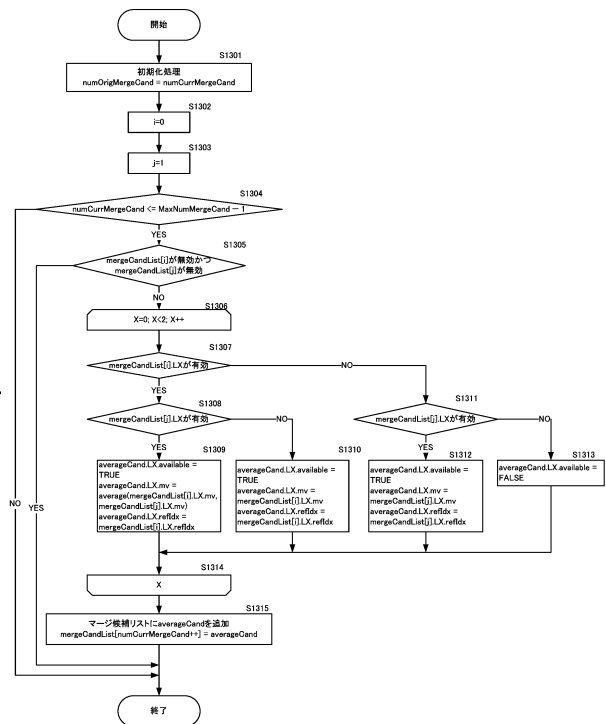
40

50

【図 6 1】



【図 6 2】



【図 6 3】

(a)

mmvd_distance_idx	MmvdDistance
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

(b)

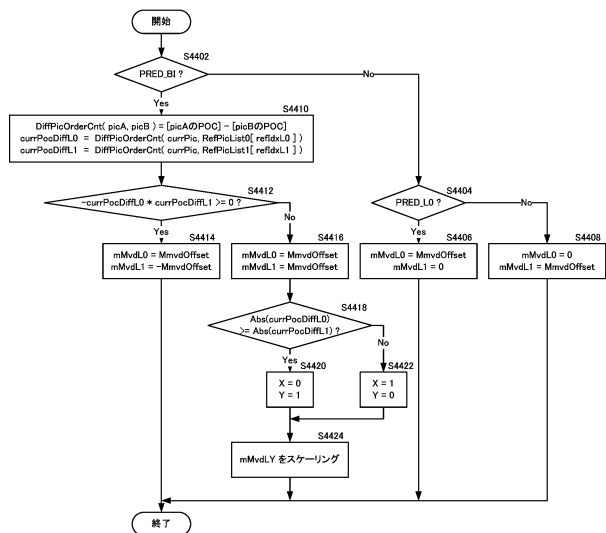
mmvd_direction_idx	MmvdSign[0]	MmvdSign[1]
0	1	0
1	-1	0
2	0	1
3	0	-1

(c)

mmvd_distance_idx	MmvdDistance
0	1/2
1	1
2	2
3	4
4	8

【図 6 4】

(a)



(b)

$td = \text{Clip3}(-128, 127, \text{currPocDiffLX})$
 $tb = \text{Clip3}(-128, 127, \text{currPocDiffLY})$
 $tx = (18384 + \text{Abs}(td) \gg 1) / td$
 $\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (tb \gg tx + 32) \gg 6)$
 $mMvdLY = \text{Clip3}(-32768, 32767, \text{Sign}(\text{distScaleFactor} * mMvdLY) * ((\text{Abs}(\text{distScaleFactor} * mMvdLY) + 127) \gg 8))$
 $\text{Clip3}(x, y, z) = \begin{cases} x & : x < x \\ y & : x > y \\ z & : \text{otherwise} \end{cases}$
 $\text{Sign}(x) = \begin{cases} 1 & : x > 0 \\ 0 & : x = 0 \\ -1 & : x < 0 \end{cases}$
 $\text{Abs}(x) = \begin{cases} x & : x \geq 0 \\ -x & : x < 0 \end{cases}$

10

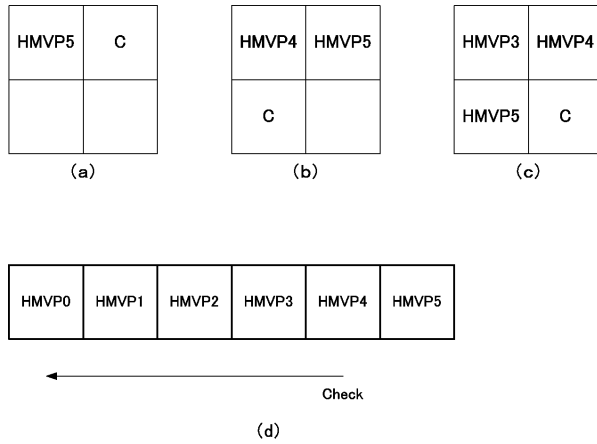
20

30

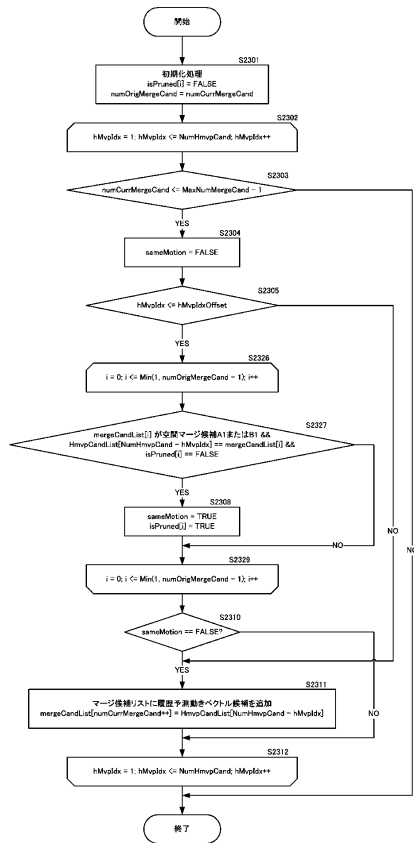
40

50

【図 6 5】



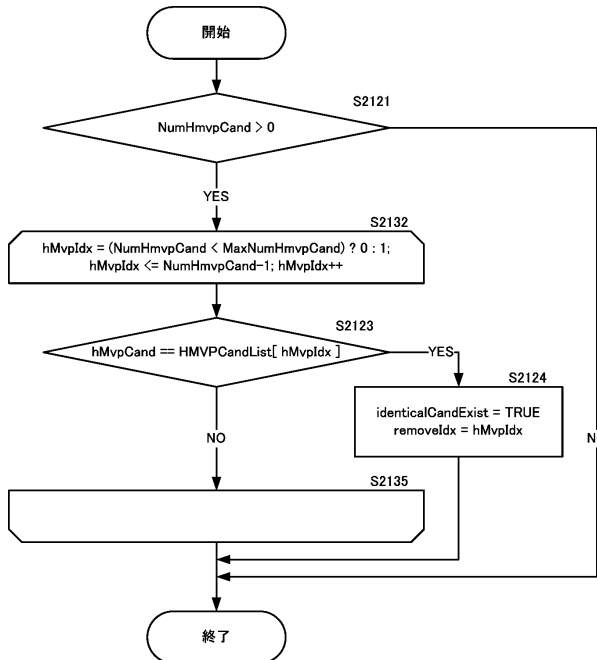
【図 6 6】



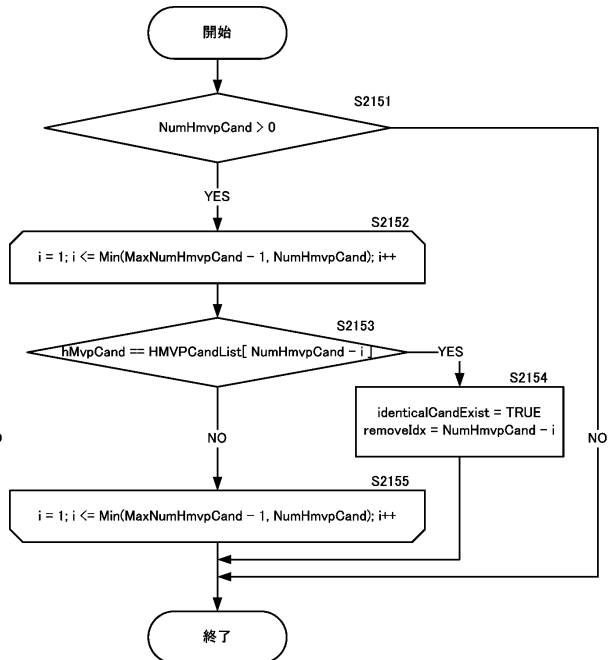
10

20

【図 6 7】



【図 6 8】

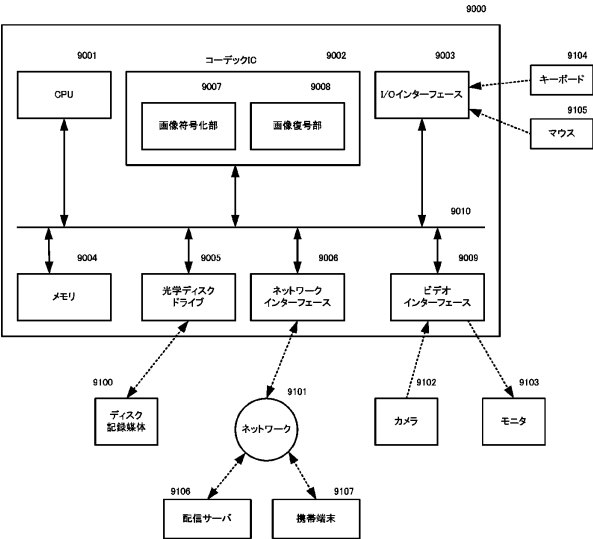


30

40

50

【図 69】



10

20

30

40

50

フロントページの続き

- 2 番地
- (72)発明者 福島 茂
神奈川県横浜市神奈川区守屋町 3 丁目 1 2 番地
- (72)発明者 竹原 英樹
神奈川県横浜市神奈川区守屋町 3 丁目 1 2 番地
- (72)発明者 坂爪 智
神奈川県横浜市神奈川区守屋町 3 丁目 1 2 番地
- (72)発明者 倉重 宏之
神奈川県横浜市神奈川区守屋町 3 丁目 1 2 番地
- 審査官 岩井 健二
- (56)参考文献 国際公開第 2 0 2 0 / 1 2 3 2 1 8 (WO , A 1)
国際公開第 2 0 2 0 / 1 2 2 6 4 0 (WO , A 1)
国際公開第 2 0 2 0 / 0 6 5 5 1 7 (WO , A 1)
国際公開第 2 0 2 0 / 0 6 0 3 7 6 (WO , A 1)
国際公開第 2 0 2 0 / 0 0 3 2 7 8 (WO , A 1)
Timofey Solovyeu, Jianle Chen, Alexander Karabutov, and Sergey Ikonin , CE4-related: History-based MVP without using the last lookup table entry , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-L0425-v6 , 12th Meeting: Macao, CN , 2018年10月 , pp.1-4
Weiwei Xu et al. , CE4-related: Constraint of Pruning in History-based Motion Vector Prediction , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-L0448-v2 , 12th Meeting: Macao, CN , 2018年10月 , pp.1-4
Hahyun Lee, et al. , Non-CE4: HMVP unification between the Merge and MVP list , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-N0373_v2 , 14th Meeting: Geneva, CH , 2019年03月 , pp.1-6
- (58)調査した分野 (Int.Cl. , D B 名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8