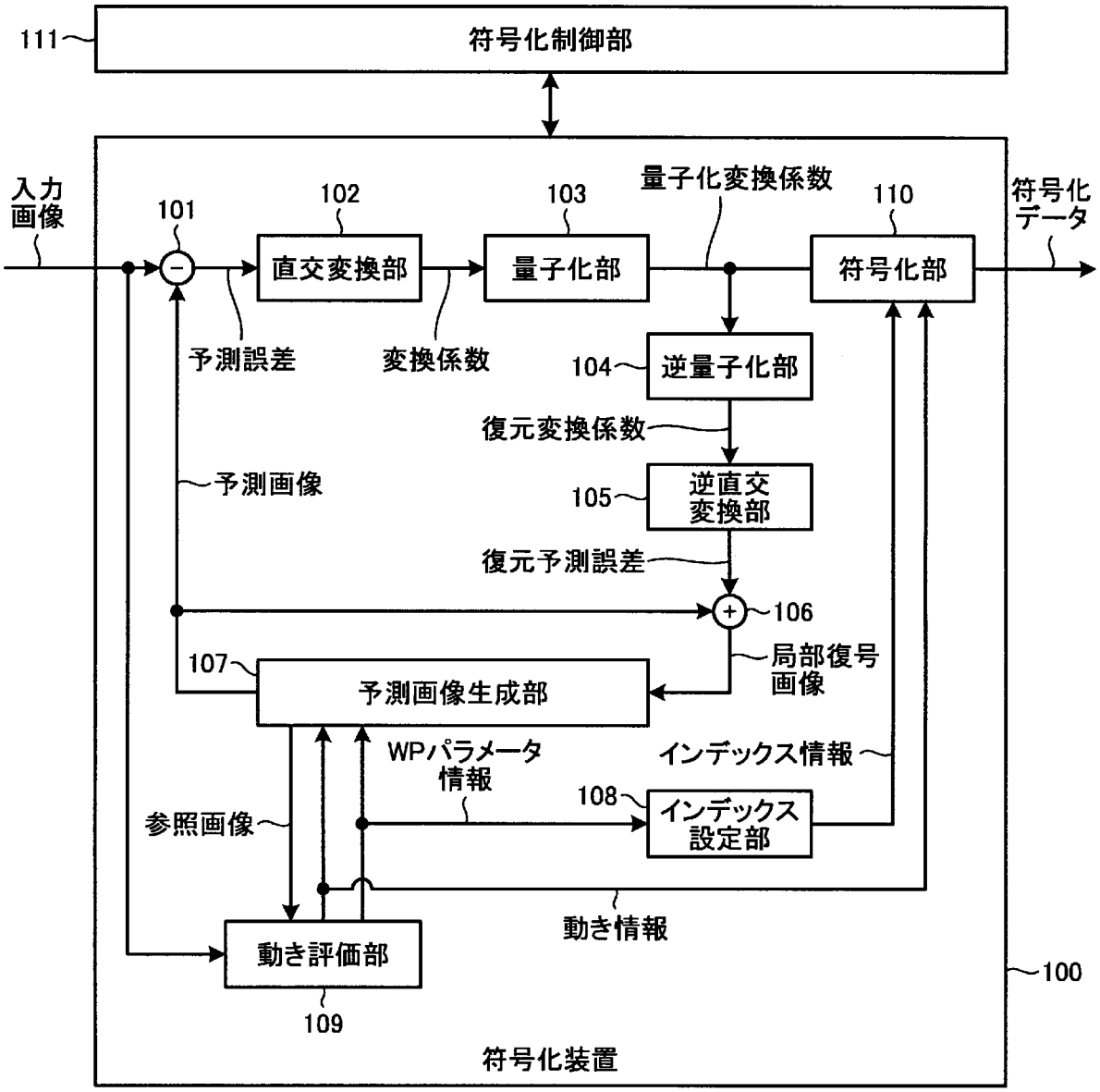
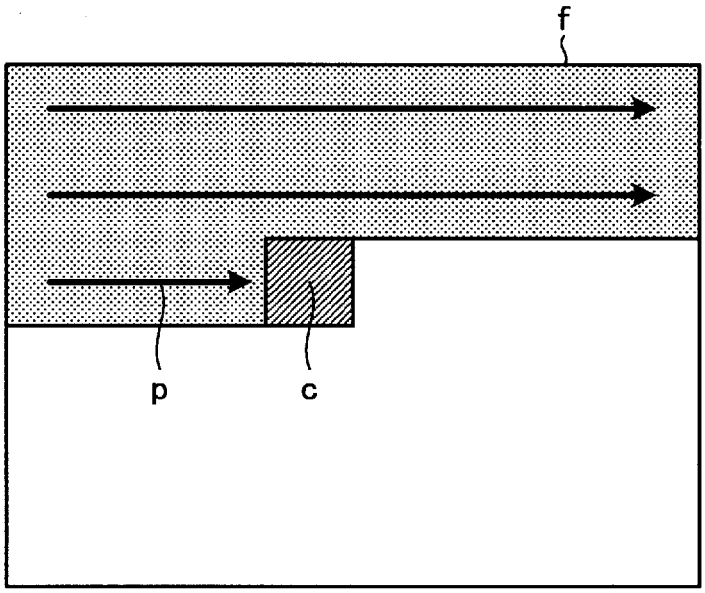
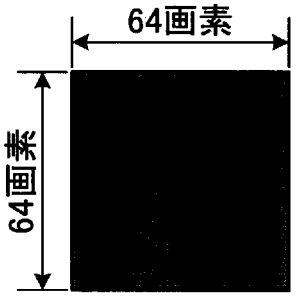
[図1]


111
符号化制御部
入力画像
101
102
直交変換部
103
量子化部
量子化変換係数
110
符号化部
符号化データ
予測誤差
変換係数
104
逆量子化部
復元変換係数
予測画像
105
逆直交変換部
復元予測誤差
106
局部復号画像
107
予測画像生成部
WPパラメータ情報
インデックス情報
108
インデックス設定部
参照画像
動き情報
動き評価部
109
100
符号化装置

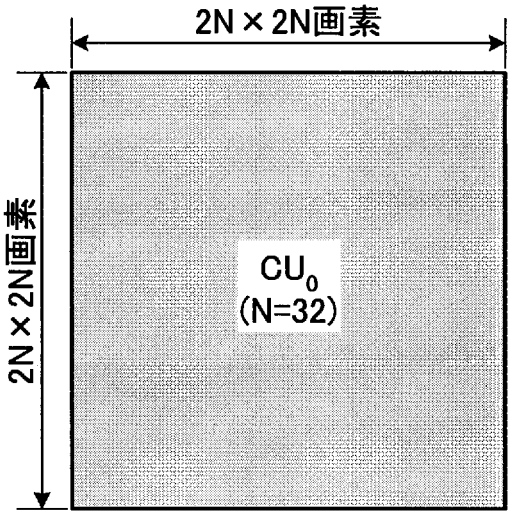
[図2]



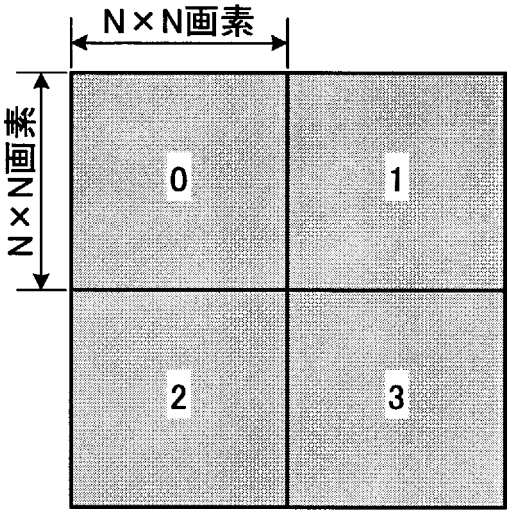
[図3A]



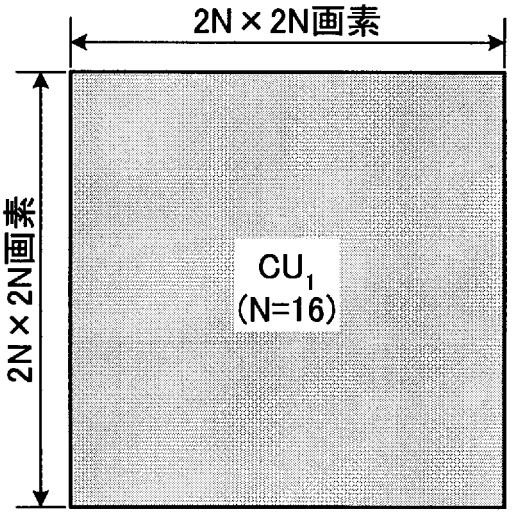
[図3B]



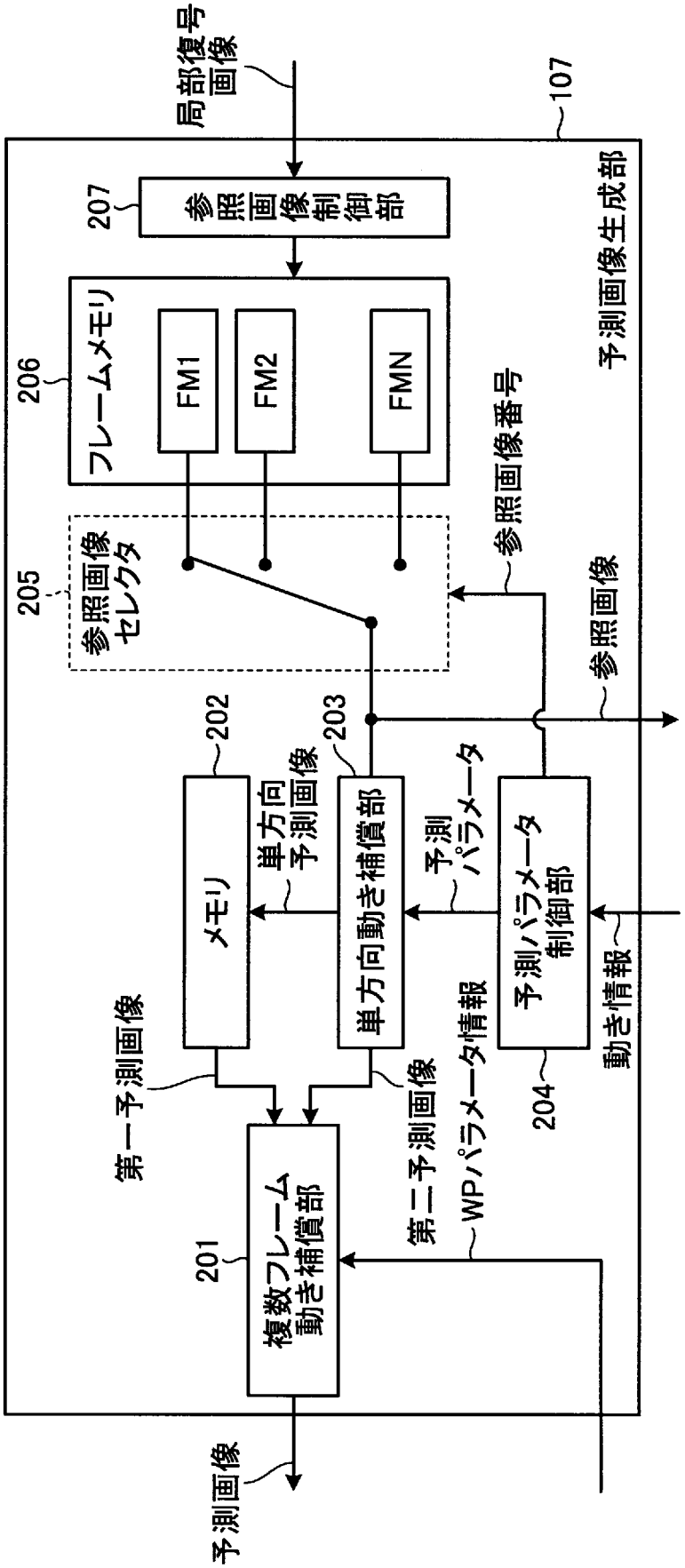
[図3C]



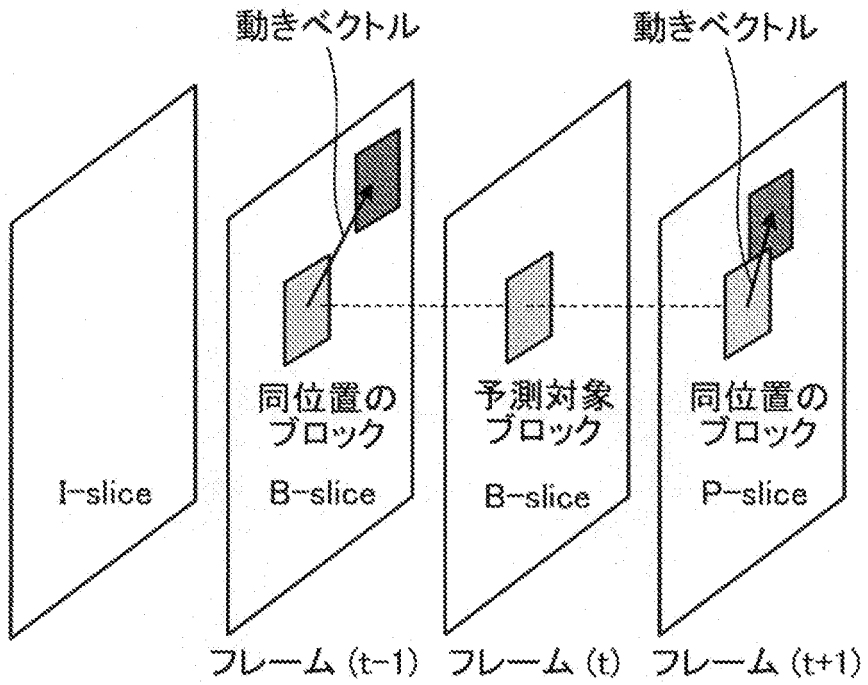
[図3D]



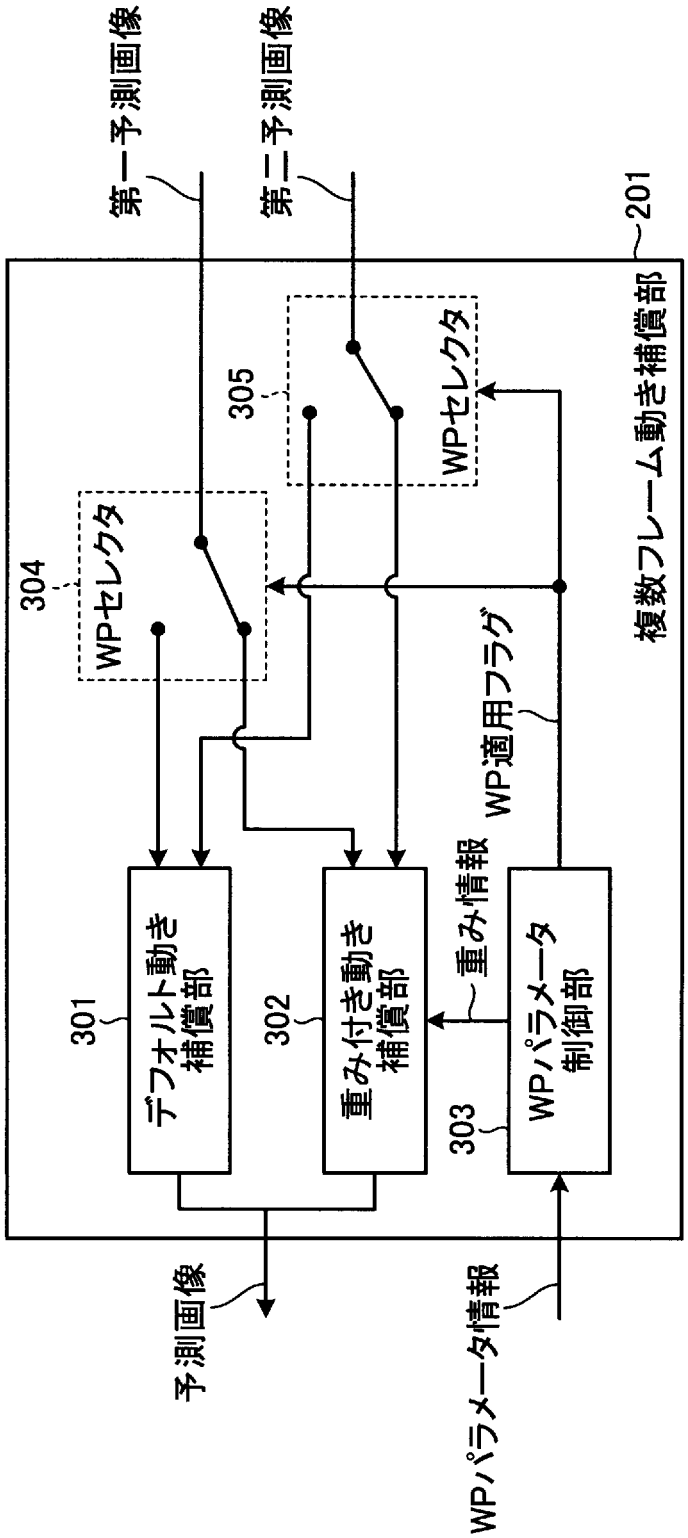
[図4]


局部復号画像
107
予測画像生成部
207
参照画像制御部
206
フレームメモリ
FM1
FM2
FMN
参照画像番号
205
参照画像セレクタ
参照画像
202
メモリ
単方向予測画像
203
単方向動き補償部
予測パラメータ
予測パラメータ制御部
動き情報
第一予測画像
第二予測画像
WPパラメータ情報
204
201
複数フレーム動き補償部
予測画像

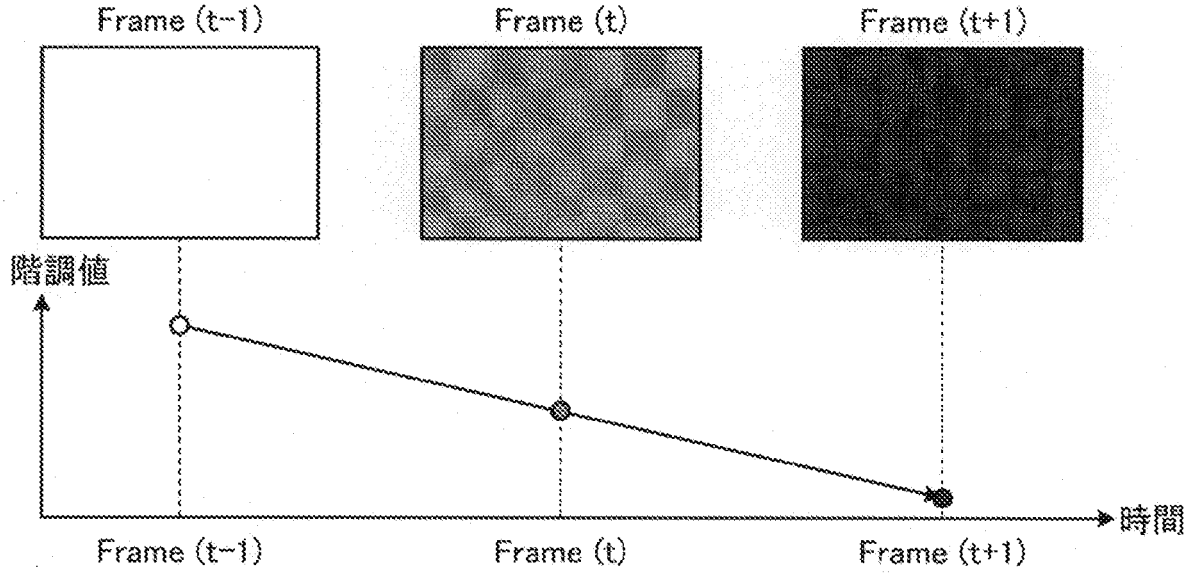
[図5]


動きベクトル
動きベクトル
同位置の
ブロック
予測対象
ブロック
同位置の
ブロック
I-slice
B-slice
B-slice
P-slice
フレーム (t-1)
フレーム (t)
フレーム (t+1)

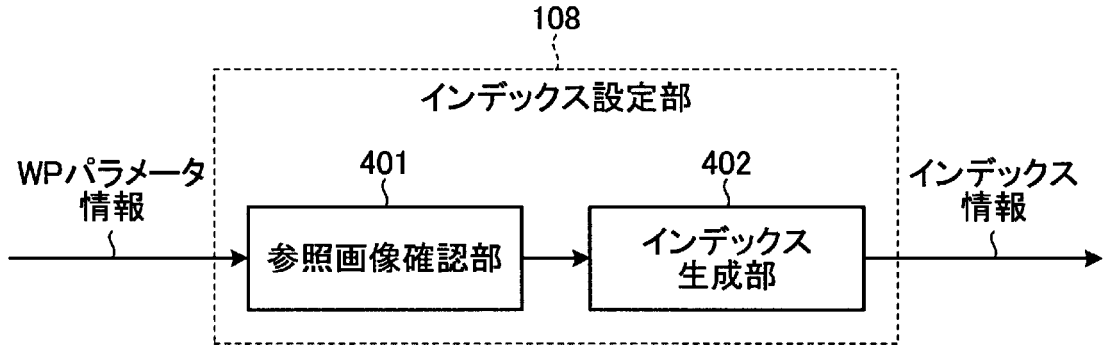
[図6]


第一予測画像
第二予測画像
201
複数フレーム動き補償部
305
WPセレクタ
304
WPセレクタ
WP適用フラグ
重み情報
301
デフォルト動き補償部
302
重み付き動き補償部
303
WPパラメータ制御部
予測画像
WPパラメータ情報

[図7]


Frame (t-1)
Frame (t)
Frame (t+1)
階調値
時間
Frame (t-1)
Frame (t)
Frame (t+1)

[図8]


108
インデックス設定部
WPパラメータ情報
401
参照画像確認部
402
インデックス生成部
インデックス情報

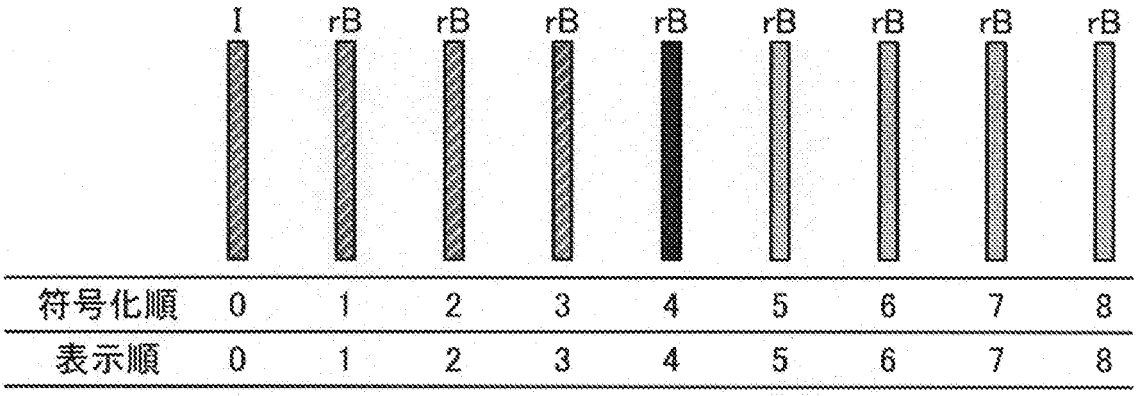
[図9A]

リスト番号	0						
参照番号	0	1	2	3	4	・・・	N
WP適用フラグ	WP_flag[0]	WP_flag[1]	WP_flag[2]	WP_flag[3]	WP_flag[4]	・・・	WP_flag[N]
重み係数	Weight[0]	Weight[1]	Weight[2]	Weight[3]	Weight[4]	・・・	Weight[N]
オフセット	Offset[0]	Offset[1]	Offset[2]	Offset[3]	Offset[4]	・・・	Offset[N]

[図9B]

リスト番号	1						
参照番号	0	1	2	3	4	・・・	N
WP適用フラグ	WP_flag[0]	WP_flag[1]	WP_flag[2]	WP_flag[3]	WP_flag[4]	・・・	WP_flag[N]
重み係数	Weight[0]	Weight[1]	Weight[2]	Weight[3]	Weight[4]	・・・	Weight[N]
オフセット	Offset[0]	Offset[1]	Offset[2]	Offset[3]	Offset[4]	・・・	Offset[N]

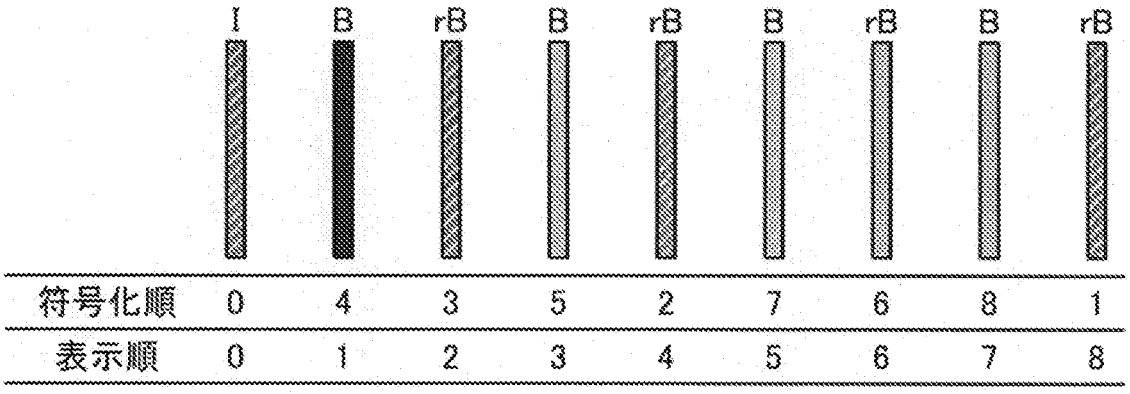
[図10]



[図11]

	参照番号(refIdx)			
	0	1	2	3
リスト0	3	2	1	0
リスト1	3	2	1	0

[図12]



[図13]

	参照番号(refIdx)			
	0	1	2	3
リスト0	0	2	4	8
リスト1	2	4	8	0

[図14]

スキャン順	参照番号(refIdx)					
	0	1	2	3	…	N
リスト0	0	2	4	6		2N
リスト1	1	3	5	7		2N+1

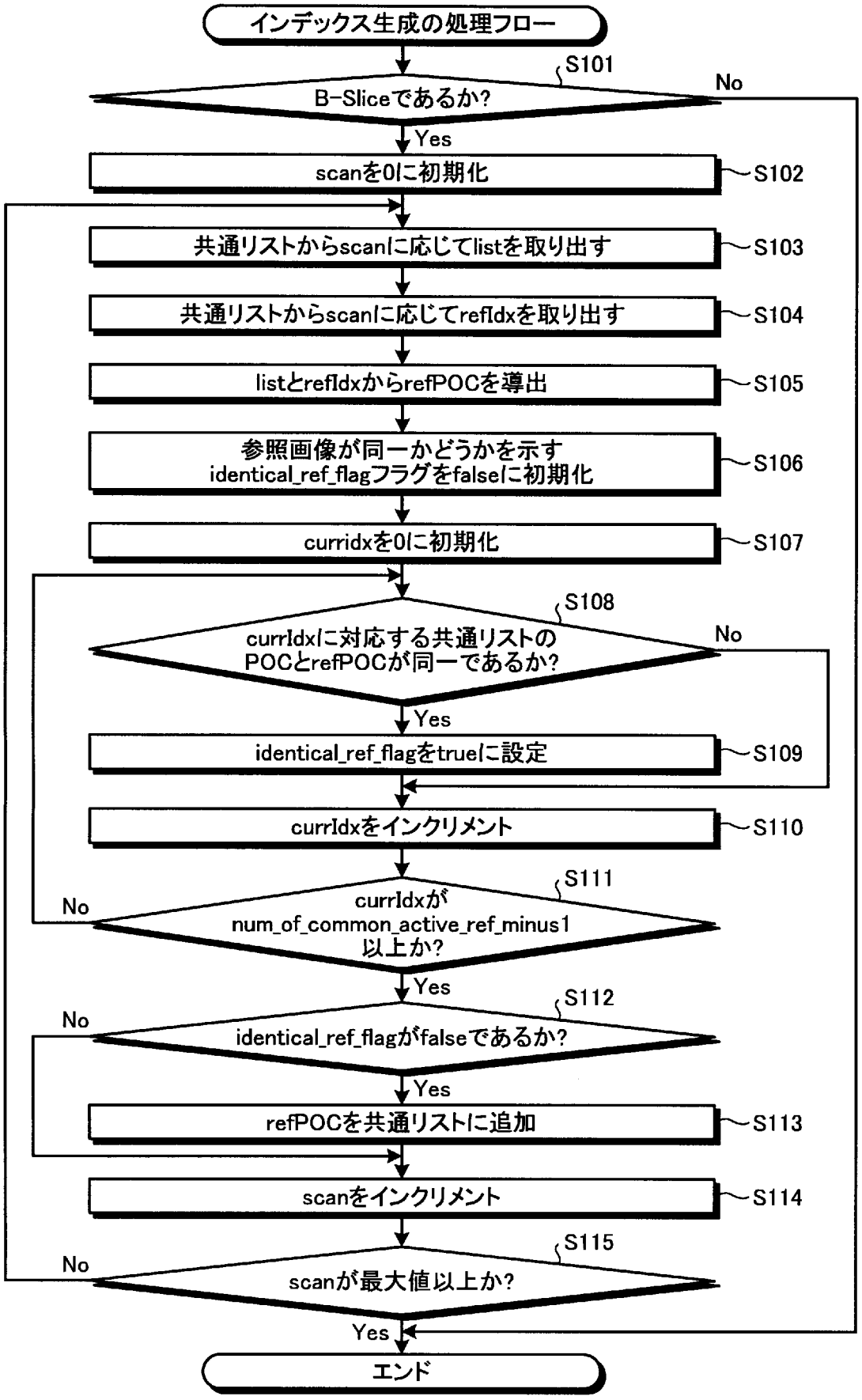
[図15]

	参照番号(refIdx)			
	0	1	2	3
共通リスト	3	2	1	0

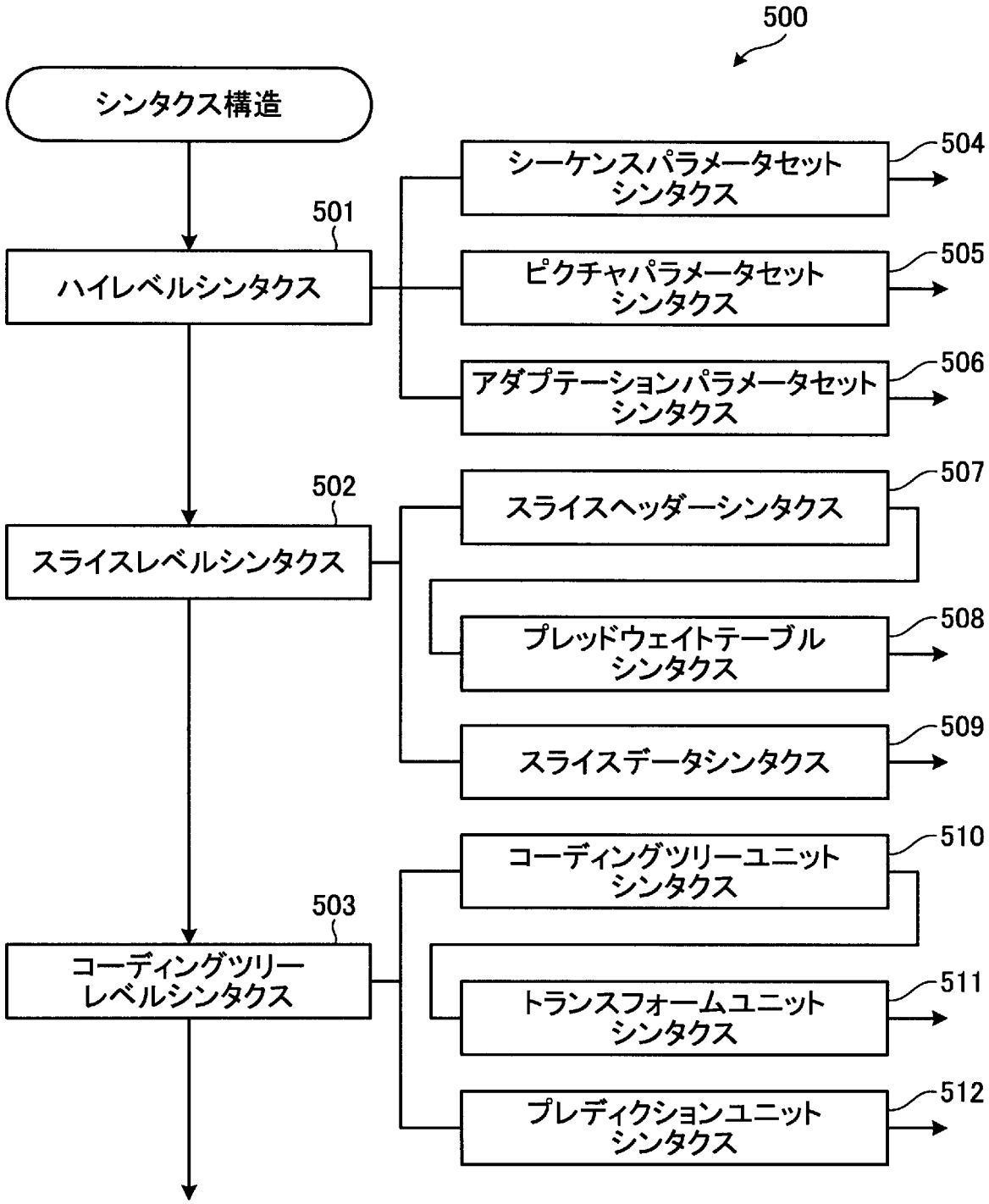
[図16]

	参照番号(refIdx)			
	0	1	2	3
共通リスト	0	2	4	8

[図17]


インデックス生成の処理フロー
S101
B-Sliceであるか?
No
Yes
scanを0に初期化
S102
共通リストからscanに応じてlistを取り出す
S103
共通リストからscanに応じてrefIdxを取り出す
S104
listとrefIdxからrefPOCを導出
S105
参照画像が同一かどうかを示す
identical_ref_flagフラグをfalseに初期化
S106
curridxを0に初期化
S107
S108
currIdxに対応する共通リストの
POCとrefPOCが同一であるか?
No
Yes
identical_ref_flagをtrueに設定
S109
currIdxをインクリメント
S110
S111
currIdxが
num_of_common_active_ref_minus1
以上か?
No
Yes
S112
identical_ref_flagがfalseであるか?
No
Yes
refPOCを共通リストに追加
S113
scanをインクリメント
S114
S115
scanが最大値以上か?
No
Yes
エンド

[図18]


500
シンタクス構造
501
ハイレベルシンタクス
502
スライスレベルシンタクス
503
コーディングツリーレベルシンタクス
シーケンスパラメータセットシンタクス
504
ピクチャパラメータセットシンタクス
505
アダプテーションパラメータセットシンタクス
506
スライスヘッダーシンタクス
507
プレッドウェイトテーブルシンタクス
508
スライスデータシンタクス
509
コーディングツリーユニットシンタクス
510
トランスフォームユニットシンタクス
511
プレディクションユニットシンタクス
512

[図19]

pic_parameter_set_rbsp( ) {	Descriptor
pic_parameter_set_id	u(8)
seq_parameter_set_id	u(8)
...	
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
weighted_pred_flag	u(1)
weighted_bipred_idc	u(2)
...	
rbsp_trailing_bits( )	
}	

[図20]

slice_header( ) {	Descriptor
...	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
...	
if(slice_type == P || slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
...	
if((weighted_pred_flag && (slice_type = = P)) || (weighted_bipred_idc == 1 && slice_type == B))	
pred_weight_table( )	
...	
}	

[図21]

pred_weight_table( ) {	Descriptor
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != MONO_IDX)	
chroma_log2_weight_denom	ue(v)
for(i = 0; i <= num_ref_common_active_minus1; i++) {	
luma_weight_common_flag	u(1)
if(luma_weight_common_flag) {	
luma_weight_common[i]	se(v)
luma_offset_common[i]	se(v)
}	
if (chroma_format_idc != MONO_IDX) {	
chroma_weight_common_flag	u(1)
if(chroma_weight_common_flag)	
for(j =0; j < COLOR_COMP − 1; j++) {	
chroma_weight_common[i][j]	se(v)
chroma_offset_common[i][j]	se(v)
}	
}	
}	

[図22]

seq_parameter_set_rbsp( ) {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
...	
num_ref_frames	ue(v)
...	
weighted_prediction_enabled_flag	u(1)
...	
rbsp_trailing_bits( )	
}	

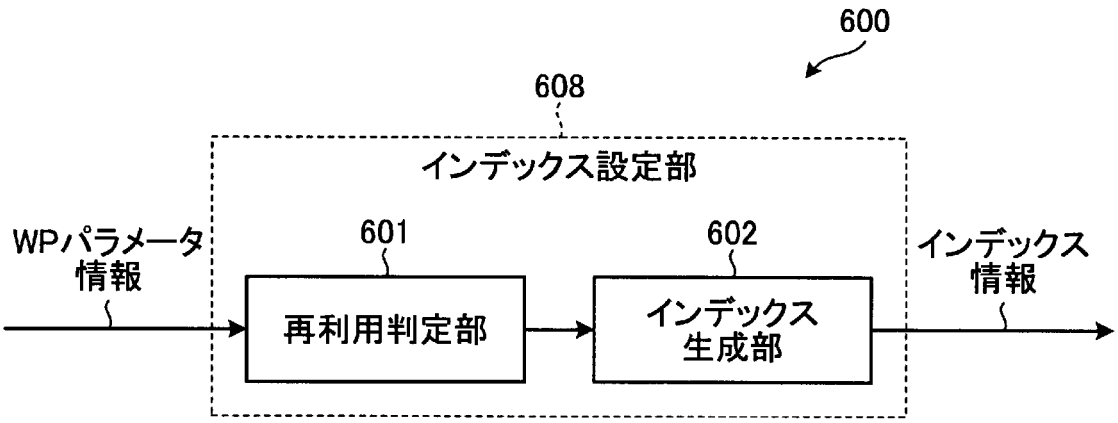
[図23]

aps_rbsp( ) {	Descriptor
aps_id	ue(v)
...	
aps_weighted_prediction_flag	u(1)
aps_weighted_bipred_idx	u(2)
...	
if(aps_weighted_prediction_flag || aps_weighted_bipred_idx = = 1)	
pred_weight_table( )	
...	
rbsp_trailing_bits( )	
}	

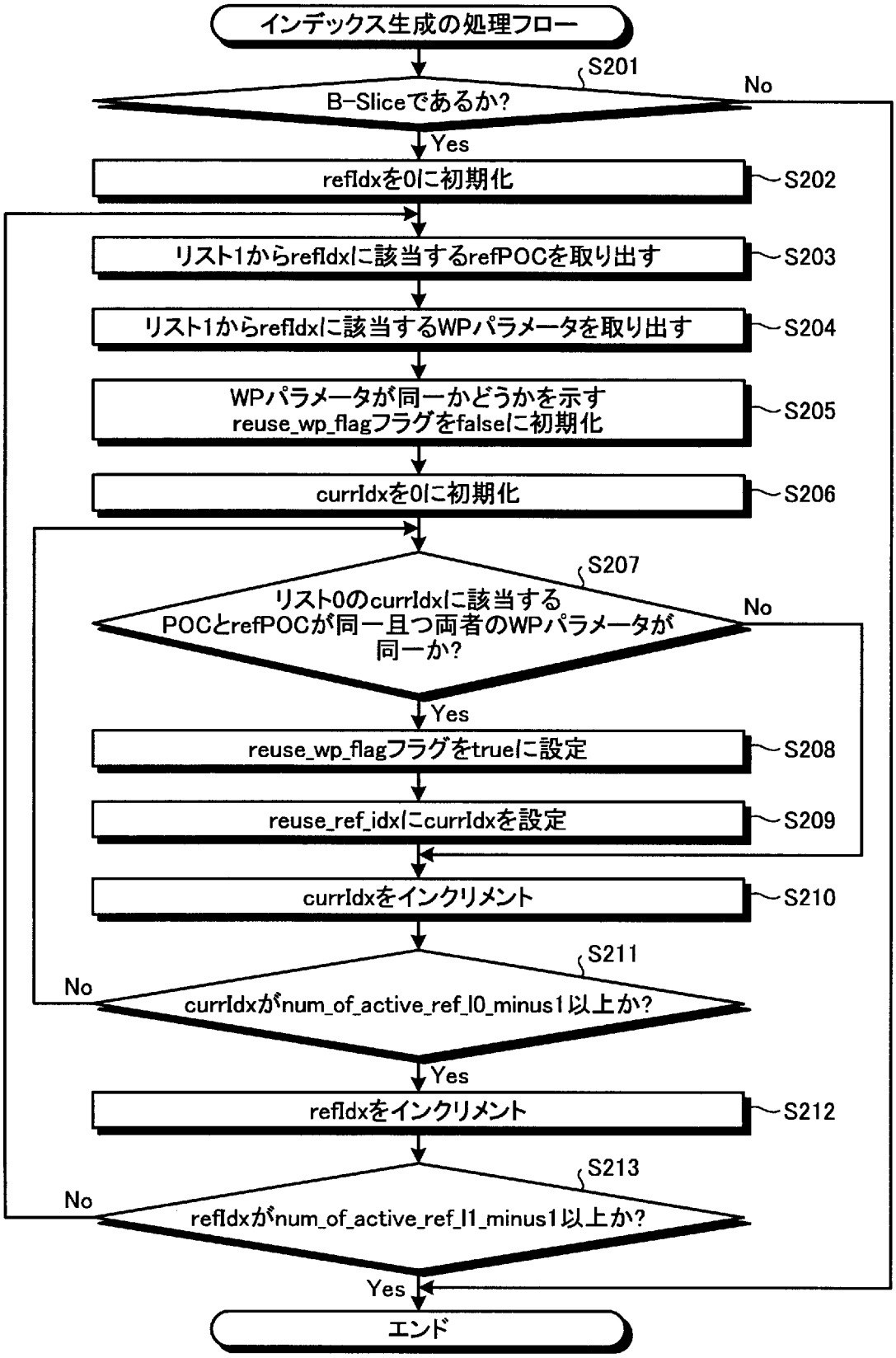
[図24]

pred_weight_table( ) {	Descriptor
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != MONO_IDX)	
chroma_log2_weight_denom	ue(v)
for(i = 0; i <= MAX_COMMON_REF_MINUS1; i++) {	
luma_weight_common_flag	u(1)
if(luma_weight_common_flag) {	
luma_weight_common[i]	se(v)
luma_offset_common[i]	se(v)
}	
if (chroma_format_idc != MONO_IDX) {	
chroma_weight_common_flag	u(1)
if(chroma_weight_common_flag)	
for(j =0; j < COLOR_COMP - 1; j++) {	
chroma_weight_common[i][j]	se(v)
chroma_offset_common[i][j]	se(v)
}	
}	
}	

[図25]


600
608
インデックス設定部
WPパラメータ情報
601
再利用判定部
602
インデックス生成部
インデックス情報

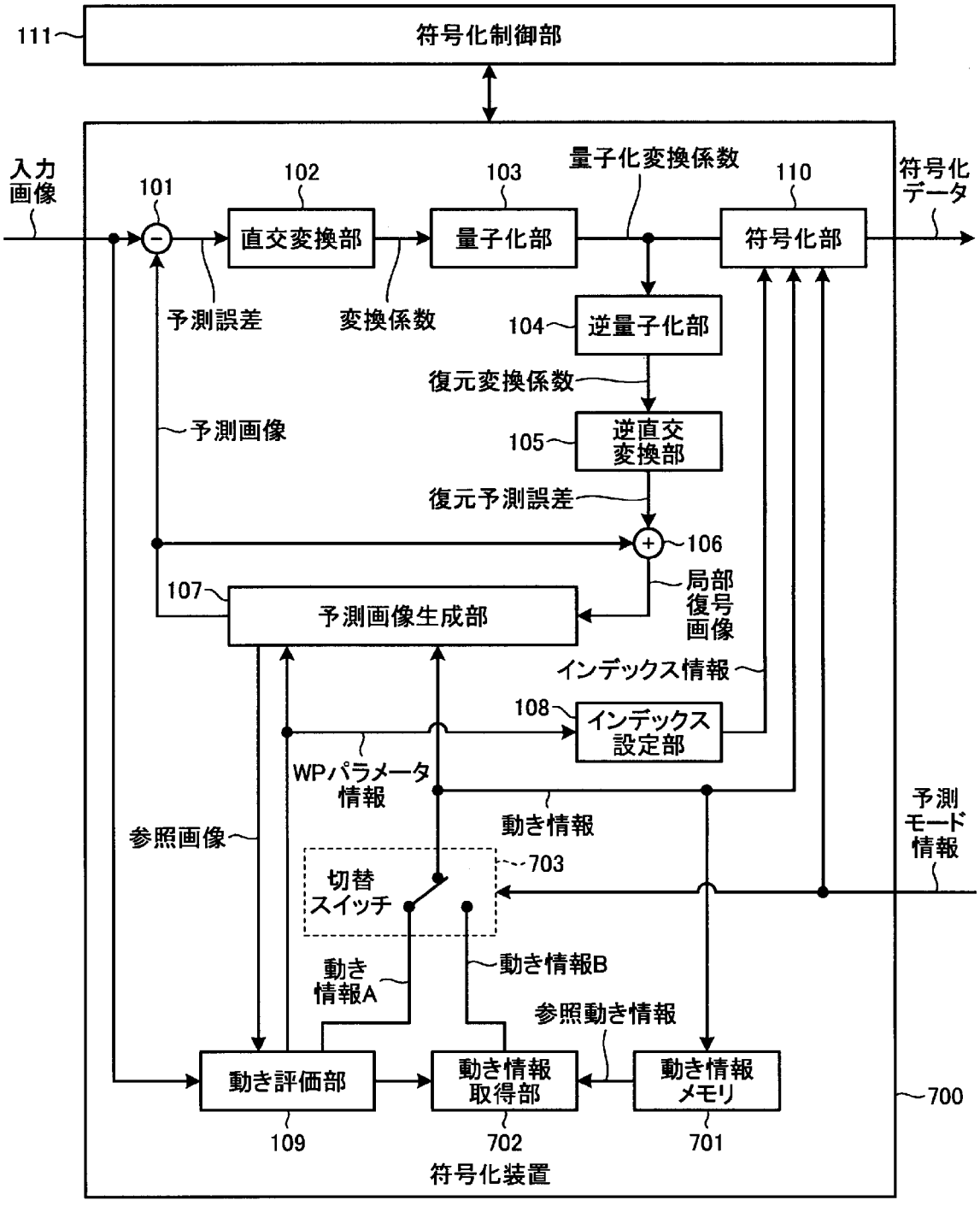
[図26]


インデックス生成の処理フロー
S201
B-Sliceであるか?
No
Yes
refIdxを0に初期化
S202
リスト1からrefIdxに該当するrefPOCを取り出す
S203
リスト1からrefIdxに該当するWPパラメータを取り出す
S204
WPパラメータが同一かどうかを示す
reuse_wp_flagフラグをfalseに初期化
S205
currIdxを0に初期化
S206
S207
リスト0のcurrIdxに該当する
POCとrefPOCが同一且つ両者のWPパラメータが
同一か?
No
Yes
reuse_wp_flagフラグをtrueに設定
S208
reuse_ref_idxにcurrIdxを設定
S209
currIdxをインクリメント
S210
S211
No
currIdxがnum_of_active_ref_l0_minus1以上か?
Yes
refIdxをインクリメント
S212
S213
No
refIdxがnum_of_active_ref_l1_minus1以上か?
Yes
エンド

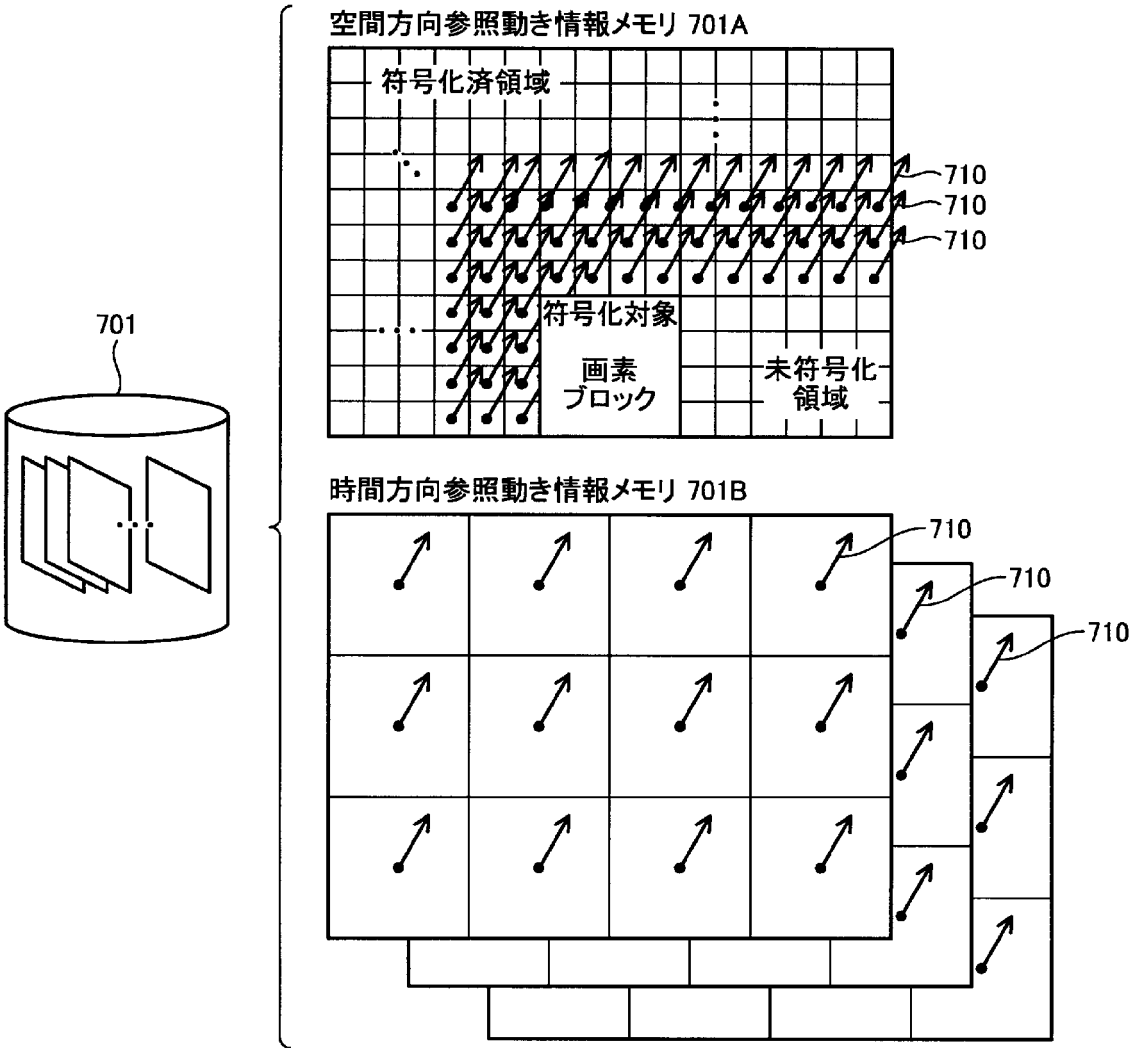
[図27]

pred_weight_table( ) {	Descriptor
luma_log2_weight_denom	ue(v)
if(chroma_format_idc != MONO_IDX)	
chroma_log2_weight_denom	ue(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {	
luma_weight_l0_flag	u(1)
if(luma_weight_l0_flag) {	
luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if (chroma_format_idc != MONO_IDX) {	
chroma_weight_l0_flag	u(1)
if(chroma_weight_l0_flag)	
for(j =0; j < COLOR_COMP - 1; j++) {	
chroma_weight_l0[i][j]	se(v)
chroma_offset_l0[i][j]	se(v)
}	
}	
}	
if(slice_type = = B_SLICE)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
reuse_wp_flag	u(1)
if(reuse_wp_flag) {	
reuse_ref_idx	se(v)
} else {	
luma_weight_l1_flag	u(1)
if(luma_weight_l1_flag) {	
luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != MONO_IDX) {	
chroma_weight_l1_flag	u(1)
if(chroma_weight_l1_flag)	
for(j = 0; j < COLOR_COMP - 1; j++) {	
chroma_weight_l1[i][j]	se(v)

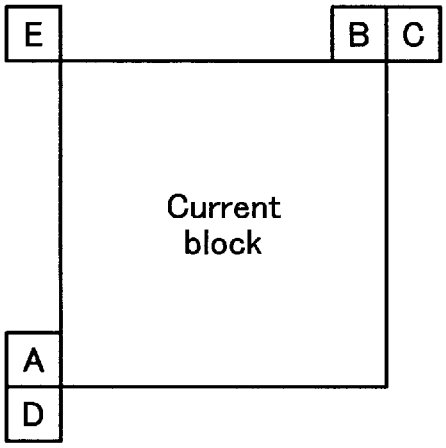
[図28]


111
符号化制御部
入力画像
101
102
直交変換部
103
量子化部
量子化変換係数
110
符号化部
符号化データ
予測誤差
変換係数
104
逆量子化部
復元変換係数
予測画像
105
逆直交変換部
復元予測誤差
106
107
予測画像生成部
局部復号画像
インデックス情報
108
インデックス設定部
WPパラメータ情報
動き情報
予測モード情報
参照画像
703
切替スイッチ
動き情報A
動き情報B
参照動き情報
動き評価部
動き情報取得部
動き情報メモリ
700
109
702
701
符号化装置

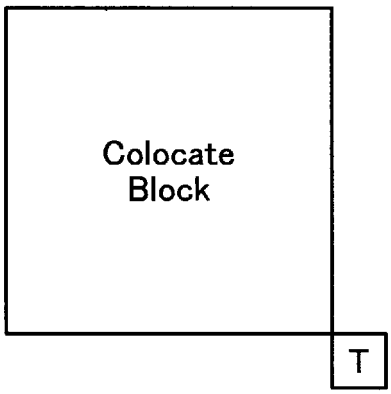
[図29]


701
空間方向参照動き情報メモリ 701A
符号化済領域
710
710
710
符号化対象
画素
ブロック
未符号化
領域
時間方向参照動き情報メモリ 701B
710
710
710

[図30A]



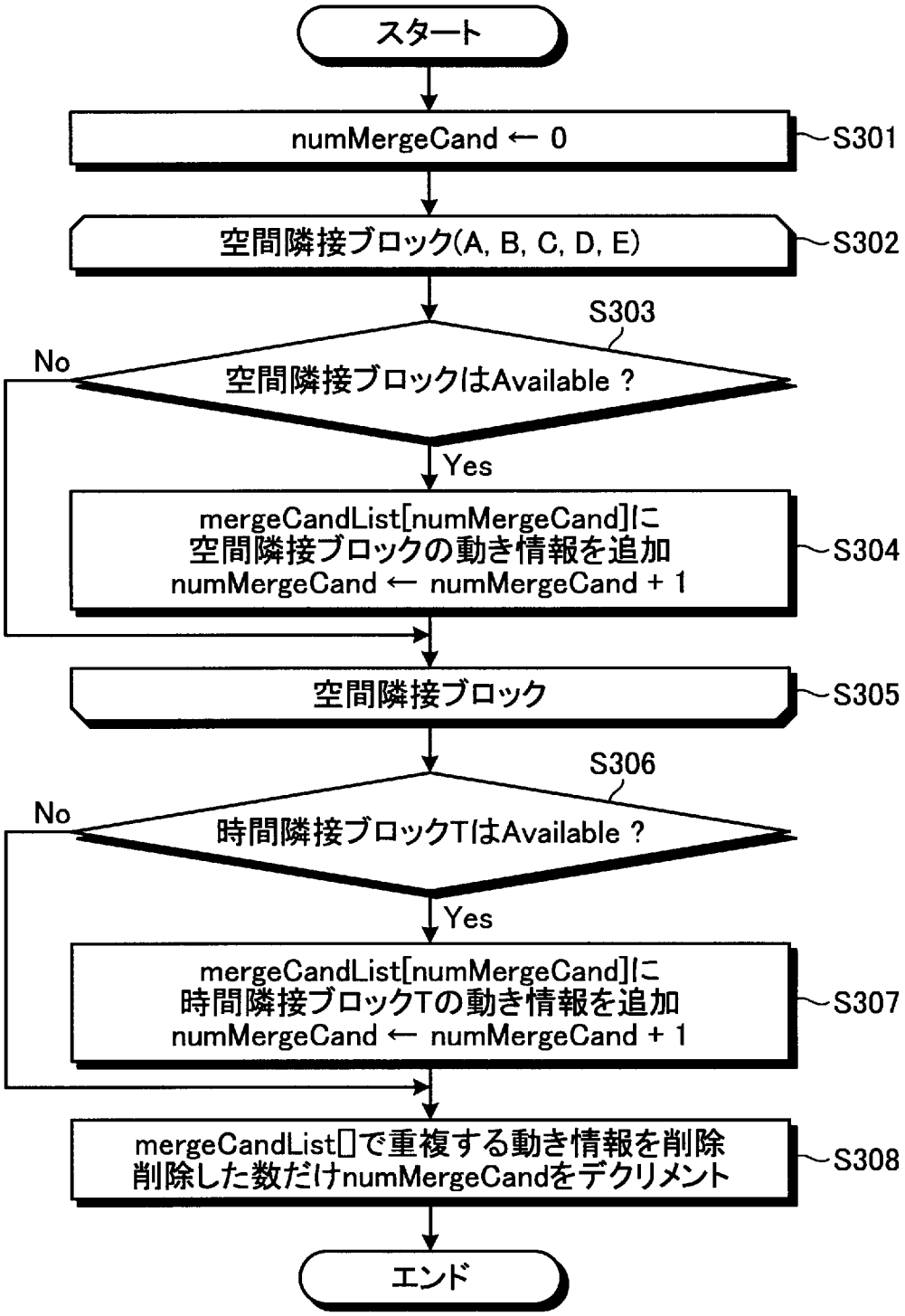
[図30B]



[図31]

画素ブロック位置	Available?	画素ブロック位置インデックス (idx)
A	YES	0
B	YES	1
C	NO	–
D	YES	2
E	YES	3
T	YES	4

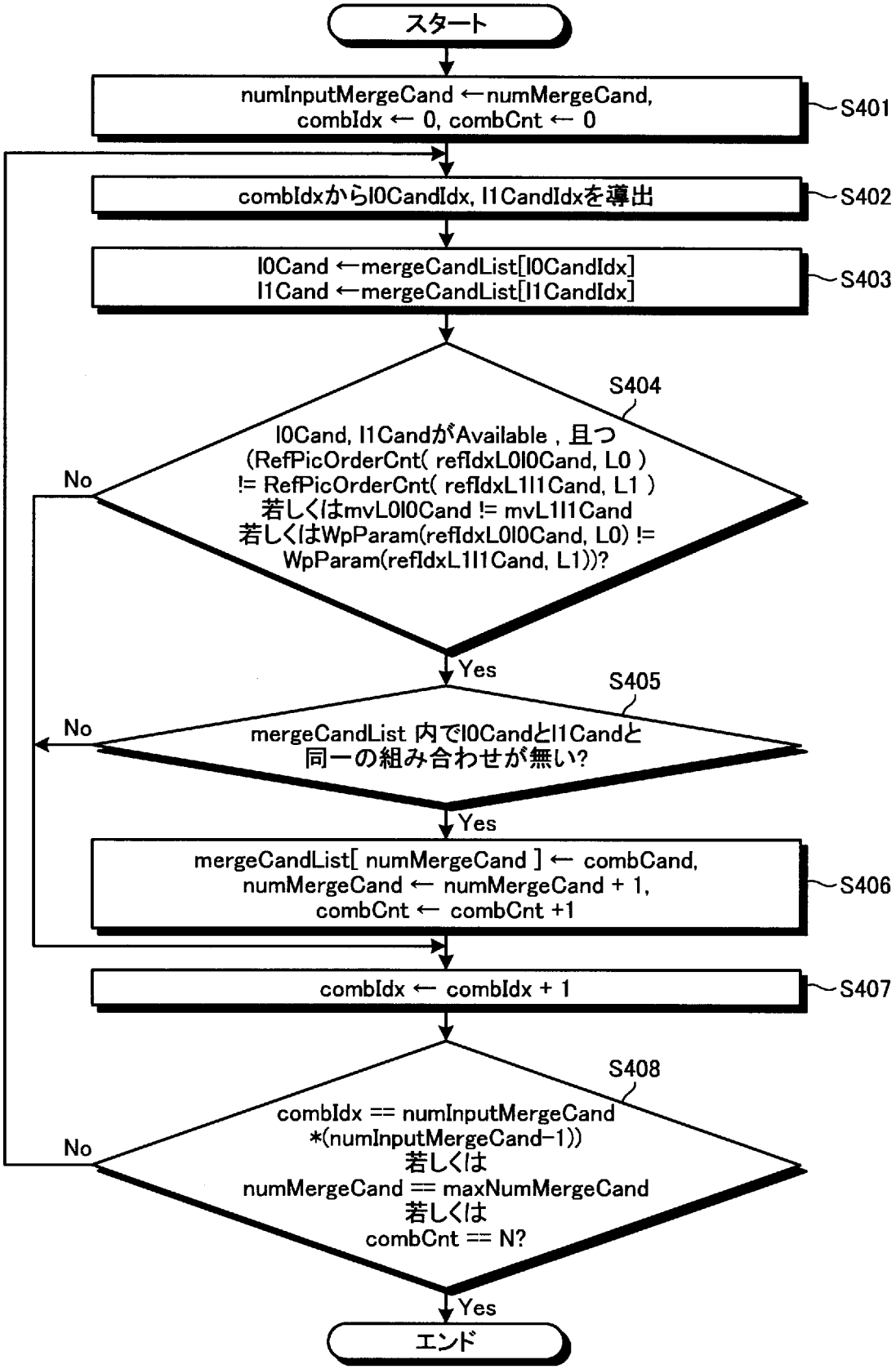
[図32]


スタート
numMergeCand ← 0
S301
空間隣接ブロック(A, B, C, D, E)
S302
S303
No
空間隣接ブロックはAvailable ?
Yes
mergeCandList[numMergeCand]に
空間隣接ブロックの動き情報を追加
numMergeCand ← numMergeCand + 1
S304
空間隣接ブロック
S305
S306
No
時間隣接ブロックTはAvailable ?
Yes
mergeCandList[numMergeCand]に
時間隣接ブロックTの動き情報を追加
numMergeCand ← numMergeCand + 1
S307
mergeCandList[]で重複する動き情報を削除
削除した数だけnumMergeCandをデクリメント
S308
エンド

[図33]

idx	MergeCandList[idx]	
	参照リスト0	参照リスト1
0	refIdx0,mv0	—
1	—	refIdx1,mv1
2	refIdx2,mv2	—
3	refIdx3,mv3	refIdx3,mv3
4	refIdx4,mv4	refIdx4,mv4

[図34]


スタート
numInputMergeCand ←numMergeCand,
combIdx ← 0, combCnt ← 0
S401
combIdxからl0CandIdx, l1CandIdxを導出
S402
l0Cand ←mergeCandList[l0CandIdx]
l1Cand ←mergeCandList[l1CandIdx]
S403
S404
l0Cand, l1CandがAvailable , 且つ
(RefPicOrderCnt(refIdxL0l0Cand, L0)
!= RefPicOrderCnt(refIdxL1l1Cand, L1)
若しくはmvL0l0Cand != mvL1l1Cand
若しくはWpParam(refIdxL0l0Cand, L0) !=
WpParam(refIdxL1l1Cand, L1))?
No
Yes
S405
mergeCandList 内でl0Candとl1Candと
同一の組み合わせが無い?
No
Yes
mergeCandList[numMergeCand] ← combCand,
numMergeCand ← numMergeCand + 1,
combCnt ← combCnt +1
S406
combIdx ← combIdx + 1
S407
S408
combIdx == numInputMergeCand
*(numInputMergeCand-1))
若しくは
numMergeCand == maxNumMergeCand
若しくは
combCnt == N?
No
Yes
エンド

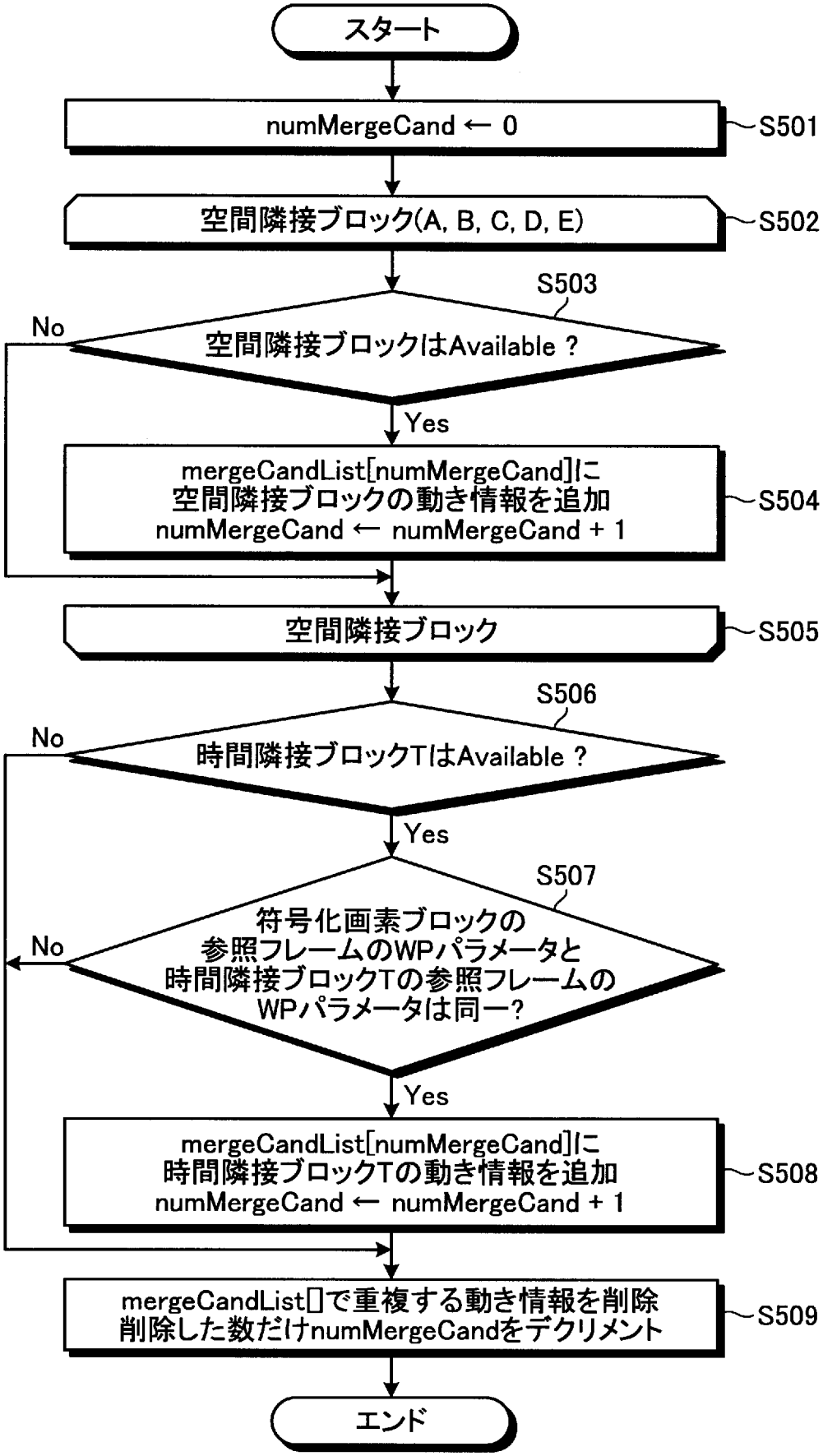
[図35]

combIdx	0	1	2	3	4	5	6	7	8	9	10	11
l0CandIdx	0	1	0	2	1	2	0	3	1	3	2	3
l1CandIdx	1	0	2	0	2	1	3	0	3	1	3	2

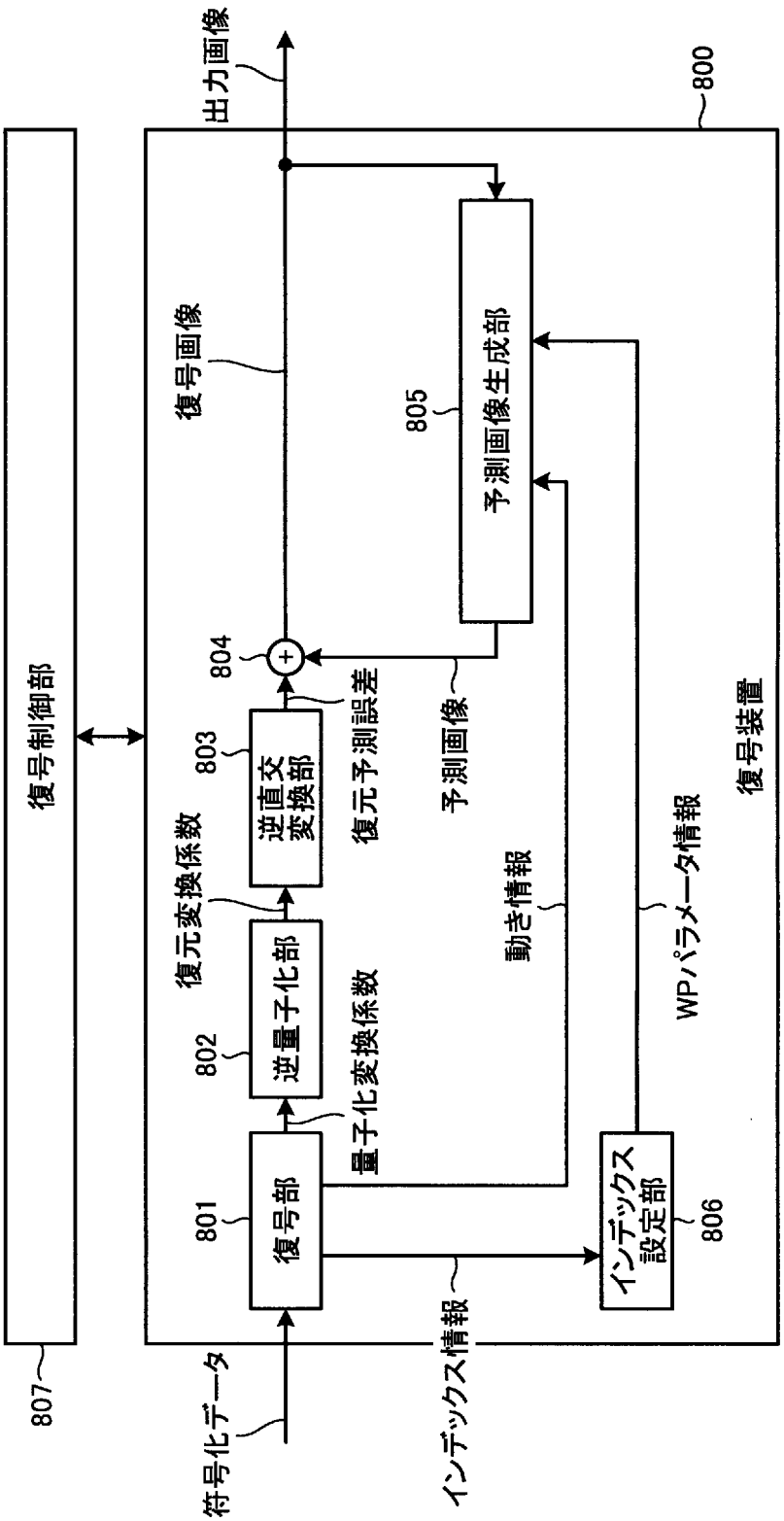
[図36]

```
prediction_unit( x0, y0 ) {
    if( skip_flag[ x0 ][ y0 ] ) {
        merge_idx[ x0 ][ y0 ]
    } else if( PredMode == MODE_INTRA ) {
        ...
    } else { /* MODE_INTER */
        if( entropy_coding_mode_flag || PartMode != PART_2Nx2N )
            merge_flag[ x0 ][ y0 ]
        if( merge_flag[ x0 ][ y0 ] ) {
            merge_idx[ x0 ][ y0 ]
        } else {
            ...
        }
    }
}
```

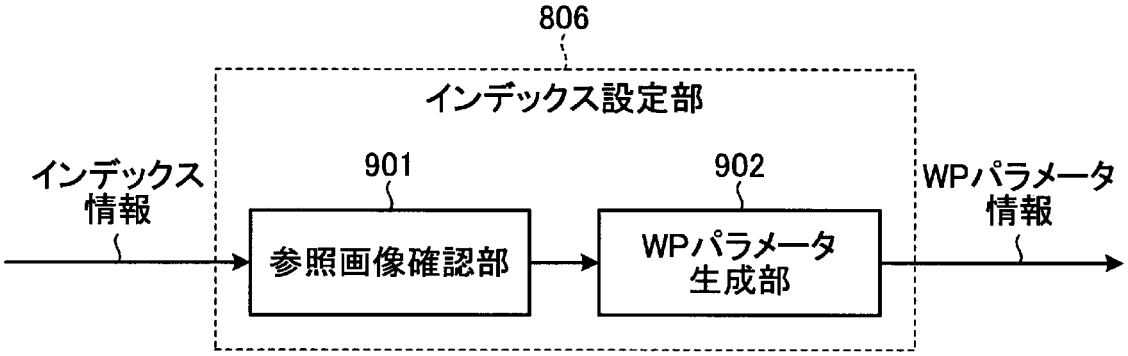
[図37]


スタート
numMergeCand ← 0
S501
空間隣接ブロック(A, B, C, D, E)
S502
S503
No
空間隣接ブロックはAvailable ?
Yes
mergeCandList[numMergeCand]に
空間隣接ブロックの動き情報を追加
numMergeCand ← numMergeCand + 1
S504
空間隣接ブロック
S505
S506
No
時間隣接ブロックTはAvailable ?
Yes
S507
No
符号化画素ブロックの
参照フレームのWPパラメータと
時間隣接ブロックTの参照フレームの
WPパラメータは同一?
Yes
mergeCandList[numMergeCand]に
時間隣接ブロックTの動き情報を追加
numMergeCand ← numMergeCand + 1
S508
mergeCandList[]で重複する動き情報を削除
削除した数だけnumMergeCandをデクリメント
S509
エンド

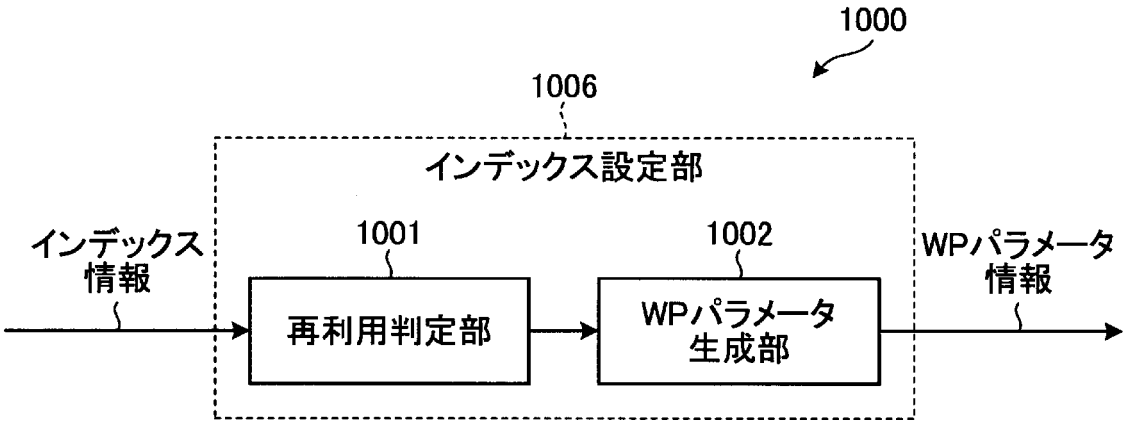
[図38]


出力画像
800
復号制御部
復号画像
805
予測画像生成部
804
803
逆直交変換部
復元予測誤差
予測画像
復元変換係数
802
逆量子化部
量子化変換係数
動き情報
801
復号部
インデックス設定部
806
WPパラメータ情報
復号装置
807
符号化データ
インデックス情報

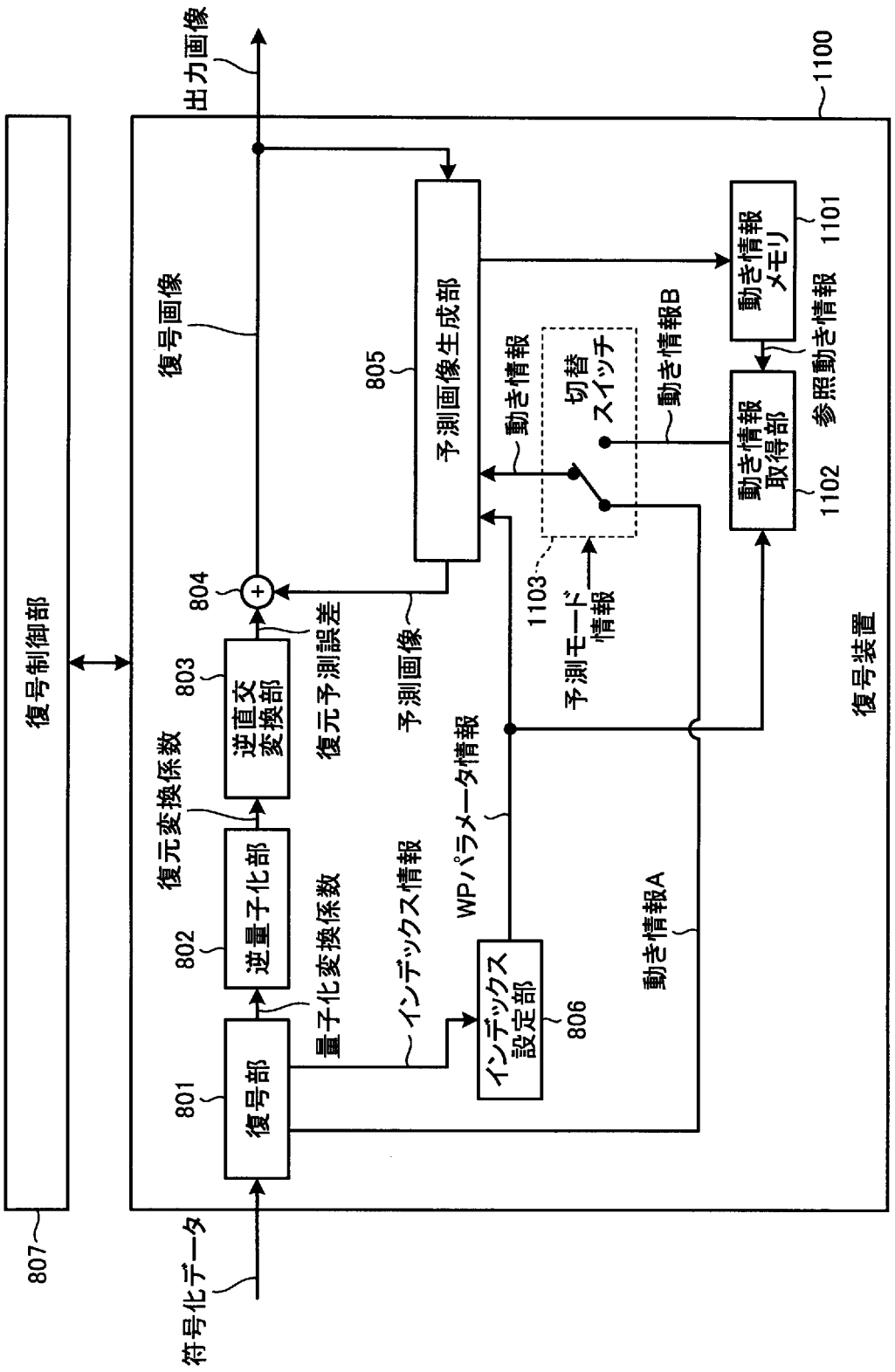
[図39]


806
インデックス設定部
インデックス
情報
901
参照画像確認部
902
WPパラメータ
生成部
WPパラメータ
情報

[図40]


1000
1006
インデックス設定部
インデックス
情報
1001
再利用判定部
1002
WPパラメータ
生成部
WPパラメータ
情報

[図41]


出力画像
1100
復号制御部
復号画像
805
予測画像生成部
動き情報
切替
スイッチ
動き情報B
動き情報
メモリ
1101
参照動き情報
動き情報
取得部
1102
804
803
逆直交
変換部
復元予測誤差
1103
予測モード
情報
予測画像
復元変換係数
802
逆量子化部
量子化変換係数
インデックス情報
WPパラメータ情報
インデックス
設定部
806
動き情報A
801
復号部
807
符号化データ
復号装置