

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3627872号
(P3627872)

(45) 発行日 平成17年3月9日(2005.3.9)

(24) 登録日 平成16年12月17日(2004.12.17)

(51) Int.C1.⁷

F 1

H04N 7/32

H04N 7/137

Z

H04N 5/21

H04N 5/21

Z

H04N 11/04

H04N 11/04

B

請求項の数 8 (全 22 頁)

(21) 出願番号

特願平7-91400

(22) 出願日

平成7年3月25日(1995.3.25)

(65) 公開番号

特開平8-88855

(43) 公開日

平成8年4月2日(1996.4.2)

審査請求日

平成14年1月10日(2002.1.10)

(31) 優先権主張番号

特願平6-191823

(32) 優先日

平成6年7月22日(1994.7.22)

(33) 優先権主張国

日本国(JP)

(73) 特許権者 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(74) 代理人 100082740

弁理士 田辺 恵基

(72) 発明者 中屋 秀雄

東京都品川区北品川6丁目7番35号ソニ
一株式会社内

(72) 発明者 近藤 哲二郎

東京都品川区北品川6丁目7番35号ソニ
一株式会社内

審査官 菅原 道晴

(56) 参考文献 特開平07-087495 (JP, A)
最終頁に続く

(54) 【発明の名称】動きベクトル検出方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

動きベクトルを検出するプロツクマッチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値から最大値及び最小値を検出する最大最小値検出ステップと、

上記最大値及び上記最小値の和の1/2の値と上記参照プロツク及び上記候補プロツクの各画素の値とを比較演算して、上記参照プロツク及び上記候補プロツクの各画素の値を値「1」又は値「0」のコード値に符号化する符号化ステップと、

上記サーチエリア内で上記候補プロツクの位置をずらし、上記コード値を用いて上記参照プロツク及び上記候補プロツク毎のマッチング演算を行い、当該演算値が最小の位置を動きベクトルとする動きベクトル検出ステップと、

直前の上記動きベクトル検出ステップで求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、上記最大最小値検出ステップ、上記符号化ステップ及び上記動きベクトル検出ステップを実行して、他の動きベクトルを算出する動きベクトル算出ステップをn回(nは0以上の整数)繰り返し、n個の動きベクトルを算出する繰り返しステップと、

上記動きベクトル検出ステップ及び上記繰り返しステップで求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、上記参照プロツク及び上記候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマッチング演算を行い、当該演算値

が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出ステップと、

上記動きベクトル検出ステップ、上記繰返しステップ及び上記画素値による動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、上記動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップと
を具えることを特徴とする動きベクトル検出方法。

【請求項 2】

上記動きベクトル検出ステップの上記マッチング演算は、上記コード値に符号化された上記参照プロック及び上記候補プロック毎の画素の値の差分の絶対値和を算出するようにした

10

ことを特徴とする請求項 1 に記載の動きベクトル検出方法。

【請求項 3】

上記動きベクトル検出ステップの上記マッチング演算は、上記コード値に符号化された上記参照プロック及び上記候補プロック毎の画素の値の排他的論理和の演算結果を積算して算出するようにした

ことを特徴とする請求項 1 に記載の動きベクトル検出方法。

【請求項 4】

動きベクトルを検出するプロックマッチングの対象となる参照プロックとサーチエリア内の全ての候補プロックとに含まれる全ての画素の値から最大値及び最小値を検出する最大最小値検出ステップと、

20

上記最大値及び上記最小値の差よりダイナミックレンジを求め、上記参照プロック及び上記候補プロックの画素値と上記最大値又は上記最小値との差分を n ビットのコード値に符号化する符号化ステップと、

上記サーチエリア内で上記候補プロックの位置をずらし、当該符号化された上記コード値の最上位桁ビットを用いて上記参照プロック及び上記候補プロック毎のマッチング演算を行い、当該演算値が最小の位置を動きベクトルとする動きベクトル検出ステップと、

直前の上記動きベクトル検出ステップで求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられた上記コード値の桁よりも一つ下の桁のビットを用いて、上記参照プロック及び上記候補プロック毎のマッチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクトル算出ステップを n 回 (n は 0 以上の整数) 繰り返す繰返しステップと、

30

上記動きベクトル検出ステップ及び上記繰返しステップで求められた動きベクトル、又は上記動きベクトル検出ステップ及び上記繰返しステップで求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、上記参照プロック及び上記候補プロックそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマッチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出ステップと、

上記動きベクトル検出ステップ、上記繰返しステップ及び上記画素値による動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、上記動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップと

40

を具えることを特徴とする動きベクトル検出方法。

【請求項 5】

入力画像データに対してラプラシアンフィルタ処理を施すラプラシアンフィルタ処理ステップと、

上記ラプラシアンフィルタ処理が施された画像から、動きベクトルを検出するプロックマッチングの対象となる参照プロックとサーチエリア内の全ての候補プロックとに含まれる画素の値の絶対値から最大値を検出する最大値検出ステップと、

上記最大値を参照プロックとサーチエリア内の全ての候補プロックのダイナミックレンジとして、各参照プロック及び各候補プロックの画素値を、極性を表す符号を含む（符号 +

50

n) ビットのコード値に符号化する符号化ステップと、
上記サーチエリア内で上記候補プロツクの位置を順次ずらし、上記コード値の符号ビット
を用いて上記参照プロツクと上記候補プロツクとのマツチング演算を行い、当該演算値が
最小の位置を第1の動きベクトルとする第1の動きベクトル検出ステップと、

上記第1の動きベクトルで動き補償を行つた後、上記サーチエリアをより小さくして、上
記コード値の最上位桁ビットを用いて、上記参照プロツク及び上記候補プロツク毎のマツ
チング演算を行い、当該演算値が最小の位置を第2の動きベクトルとする第2の動きベク
トル検出ステップと、

直前の上記第2の動きベクトル検出ステップで求められた第2の動きベクトルで動き補償
を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられた上記コー
ド値の桁よりも一つ下の桁のビットを用いて、上記参照プロツク及び上記候補プロツク毎
のマツチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクト
ル算出ステップをn回（nは0以上の整数）繰り返す繰返しステップと、

上記第1、第2動きベクトル検出ステップ及び上記繰返しステップで求められた動きベク
トル、又は上記第1、第2の動きベクトル検出ステップ及び上記繰返しステップで求め
られた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、上
記参照プロツク及び上記候補プロツクそれぞれの画素値を用いて、最後に求められたサー
チエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング
演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動
きベクトル検出ステップと、

上記第1、第2の動きベクトル検出ステップ、上記繰返しステップ及び上記画素値によ
る動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、当該動きベクト
ルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップと
を具えることを特徴とする動きベクトル検出方法。

【請求項6】

動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内
の全ての候補プロツクとに含まれる画素の値から最大値及び最小値を検出する最大最小値
検出手段と、

上記最大値及び上記最小値の和の1/2の値と上記参照プロツク及び上記候補プロツクの
各画素の値とを比較演算して、上記参照プロツク及び上記候補プロツクの各画素の値を値
「1」又は値「0」のコード値に符号化する符号化手段と、

上記サーチエリア内で上記候補プロツクの位置をずらし、上記コード値を用いて上記参照
プロツク及び上記候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を動
きベクトルとする動きベクトル検出手段と、

直前の上記動きベクトル検出手段で求められた動きベクトルで動き補償を行つた後、直前
のサーチエリアより小さいサーチエリアで、上記最大最小値検出手段、上記符号化手段及
び上記動きベクトル検出手段による処理を実行して、他の動きベクトルを算出する動きベ
クトル算出手段による処理をn回（nは0以上の整数）繰り返し、n個の動きベクトルを
算出する繰返し手段と、

上記動きベクトル検出手段及び上記繰返し手段により求められた動きベクトルのうち、最
後に求められた動きベクトルに応じて動き補償を行つた後、上記参照プロツク及び上記候
補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、
所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が
最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出手段と、

上記動きベクトル検出手段、上記繰返し手段及び上記画素値による動きベクトル検出手段
により求められた複数の動きベクトルの和を求め、上記動きベクトルの和を最終的な動き
ベクトルとして出力する動きベクトル出力手段と
を具えることを特徴とする動きベクトル検出装置。

【請求項7】

動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内

10

20

30

40

50

の全ての候補プロツクとに含まれる全ての画素の値から最大値及び最小値を検出する最大最小値検出手段と、

上記最大値及び上記最小値の差よりダイナミックレンジを求め、上記参照プロツク及び上記候補プロツクの画素値と上記最大値又は上記最小値との差分をnビットのコード値に符号化する符号化手段と、

上記サーチエリア内で上記候補プロツクの位置をずらし、当該符号化された上記コード値の最上位桁ビットを用いて上記参照プロツク及び上記候補プロツク毎のマッチング演算を行い、当該演算値が最小の位置を動きベクトルとする動きベクトル検出手段と、

直前の上記動きベクトル検出手段により求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられた上記コード値の桁よりも一つ下の桁のビットを用いて、上記参照プロツク及び上記候補プロツク毎のマッチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクトル算出手段による処理をn回(nは0以上の整数)繰り返す繰返し手段と、

上記動きベクトル検出手段及び上記繰返し手段により求められた動きベクトル、又は上記動きベクトル検出手段及び上記繰返し手段により求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、上記参照プロツク及び上記候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマッチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出手段と、

上記動きベクトル検出手段、上記繰返し手段及び上記画素値による動きベクトル検出手段により求められた複数の動きベクトルの和を求め、上記動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力手段と
を具えることを特徴とする動きベクトル検出装置。

【請求項8】

入力画像データに対してラプラシアンフィルタ処理を施すラプラシアンフィルタと、上記ラプラシアンフィルタ処理が施された画像から、動きベクトルを検出するプロツクマッチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値の絶対値から最大値を検出する最大値検出手段と、

上記最大値を参照プロツクとサーチエリア内の全ての候補プロツクのダイナミックレンジとして、各参照プロツク及び各候補プロツクの画素値を、極性を表す符号を含む(符号+n)ビットのコード値に符号化する符号化手段と、

上記サーチエリア内で上記候補プロツクの位置を順次ずらし、上記コード値の符号ビットを用いて上記参照プロツクと上記候補プロツクとのマッチング演算を行い、当該演算値が最小の位置を第1の動きベクトルとする第1の動きベクトル検出手段と、

上記第1の動きベクトルで動き補償を行つた後、上記サーチエリアをより小さくして、上記コード値の最上位桁ビットを用いて、上記参照プロツク及び上記候補プロツク毎のマッチング演算を行い、当該演算値が最小の位置を第2の動きベクトルとする第2の動きベクトル検出手段と、

直前の上記第2の動きベクトル検出手段で求められた第2の動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられた上記コード値の桁よりも一つ下の桁のビットを用いて、上記参照プロツク及び上記候補プロツク毎のマッチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクトル算出手段による処理をn回(nは0以上の整数)繰り返す繰返し手段と、

上記第1、第2動きベクトル検出手段及び上記繰返し手段により求められた動きベクトル、又は上記第1、第2の動きベクトル検出手段及び上記繰返し手段で求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、上記参照プロツク及び上記候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマッチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出手段と、

10

20

30

40

50

上記第1、第2の動きベクトル検出手段、上記繰返し手段及び上記画素値による動きベクトル検出手段で求められた複数の動きベクトルの和を求め、上記動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力手段とを具えることを特徴とする動きベクトル検出装置。

【発明の詳細な説明】

【0001】

【目次】

以下の順序で本発明を説明する。

産業上の利用分野

従来の技術

10

発明が解決しようとする課題

課題を解決するための手段

作用

実施例

(1) 動きベクトル検出の原理(図1)

(2) 第1実施例の動きベクトル検出方法及び装置(図2～図4)

(3) 第2実施例の動きベクトル検出方法及び装置(図2、図5及び図6)

(4) 第3実施例の動きベクトル検出方法及び装置(図7～図11)

(5) 他の実施例

発明の効果

20

【0002】

【産業上の利用分野】

本発明は動きベクトル検出方法及び装置に関し、特に時間的に異なる2つの画像データを用いて画像の動きベクトルを検出するものに適用し得る。

【0003】

【従来の技術】

従来、動画像の処理として、動きベクトルすなわち時間的に異なる画像中の物体の動き方向と大きさや速さを用いるものがある。例えば画像の高能率符号化における動き補償フレーム間符号化や、フレーム間時間領域フィルタによるテレビジョン雑音低減装置における動きによるパラメータ制御等に、動きベクトルが用いられる。この画像の動きベクトルを求める動きベクトル検出方法として、プロツクマッチング法がある(特公昭54-124927号公報)。

30

【0004】

このプロツクマッチング法では、まず1つの画面を適当な数画素からなるプロツクに分割する。続いてこのようにプロツク化された画像データを参照プロツクとし、当該参照プロツクが動いた領域を検索するために、時間的に異なる画面の対応する領域を含むようにプロツク化した画像データを候補プロツクとする。この参照プロツクと候補プロツクとのマッチング演算を、サーチ領域内で1画素ずつずらしながら行い、評価値が最小となる最適値を求ることにより、参照プロツクの動きベクトルを検出する。

40

【0005】

【発明が解決しようとする課題】

ところがプロツクマッチング法で動きベクトルを検出する際には、検出対象のプロツクの全ての画素に対して、検出範囲となる全てのサーチ領域をくまなくサーチし、その差分を求める必要がある。このため動きベクトルを検出する計算量が大きくなり、装置自体が大型化したり演算時間が長くなる問題があつた。

【0006】

本発明は以上の点を考慮してなされたもので、簡易な回路構成で階層的にレベル方向で分解能を上げて精度良く動きベクトルを検出し得る動きベクトル検出方法及び装置を提案しようとするものである。

【0007】

50

【課題を解決するための手段】

かかる課題を解決するため本発明においては、動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値から最大値及び最小値を検出する最大最小値検出ステップと、最大値及び最小値の和の 1 / 2 の値と参照プロツク及び候補プロツクの各画素の値とを比較演算して、参照プロツク及び候補プロツクの各画素の値を値「1」又は値「0」のコード値に符号化する符号化ステップと、サーチエリア内で候補プロツクの位置をずらし、コード値を用いて参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を動きベクトルとする動きベクトル検出ステップと、直前の動きベクトル検出ステップで求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、最大最小値検出ステップ、符号化ステップ及び動きベクトル検出ステップを実行して、他の動きベクトルを算出する動きベクトル算出ステップを n 回 (n は 0 以上の整数) 繰り返し、n 個の動きベクトルを算出する繰返しステップと、動きベクトル検出ステップ及び繰返しステップで求められた動きベクトルのうち、最後に求められた動きベクトルに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出ステップと、動きベクトル検出ステップ、繰返しステップ及び画素値による動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップとを設けるようにする。
10

【0008】

また本発明においては、動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる全ての画素の値から最大値及び最小値を検出する最大最小値検出ステップと、最大値及び最小値の差よりダイナミックレンジを求め、参照プロツク及び候補プロツクの画素値と最大値又は最小値との差分を n ビットのコード値に符号化する符号化ステップと、サーチエリア内で候補プロツクの位置をずらし、当該符号化されたコード値の最上位桁ビットを用いて参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を動きベクトルとする動きベクトル検出ステップと、直前の動きベクトル検出ステップで求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられたコード値の桁よりも一つ下の桁のビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクトル算出ステップを n 回 (n は 0 以上の整数) 繰り返す繰返しステップと、動きベクトル検出ステップ及び繰返しステップで求められた動きベクトル、又は動きベクトル検出ステップ及び繰返しステップで求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出ステップと、動きベクトル検出ステップ、繰返しステップ及び画素値による動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップとを設けるようにする。
20

【0009】

さらに本発明においては、入力画像データに対してラプラシアンフィルタ処理を施すラプラシアンフィルタ処理ステップと、ラプラシアンフィルタ処理が施された画像から、動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値の絶対値から最大値を検出する最大値検出ステップと、最大値を参照プロツクとサーチエリア内の全ての候補プロツクのダイナミックレンジとして、各参照プロツク及び各候補プロツクの画素値を、極性を表す符号を含む (符号 40

+ n) ビットのコード値に符号化する符号化ステップと、サーチエリア内で候補プロツクの位置を順次ずらし、コード値の符号ビットを用いて参照プロツクと候補プロツクとのマツチング演算を行い、当該演算値が最小の位置を第 1 の動きベクトルとする第 1 の動きベクトル検出ステップと、第 1 の動きベクトルで動き補償を行つた後、サーチエリアをより小さくして、コード値の最上位桁ビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を第 2 の動きベクトルとする第 2 の動きベクトル検出ステップと、直前の第 2 の動きベクトル検出ステップで求められた第 2 の動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられたコード値の桁よりも一つ下の桁のビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする動きベクトル算出ステップを n 回 (n は 0 以上の整数) 繰り返す繰返しステップと、第 1 、第 2 動きベクトル検出ステップ及び繰返しステップで求められた動きベクトル、又は第 1 、第 2 の動きベクトル検出ステップ及び繰返しステップで求められた動きベクトルのうち最後に求められた動きベクトルに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとする画素値による動きベクトル検出ステップと、第 1 、第 2 の動きベクトル検出ステップ、繰返しステップ及び画素値による動きベクトル検出ステップで求められた複数の動きベクトルの和を求め、動きベクトルの和を最終的な動きベクトルとして出力する動きベクトル出力ステップとを設けるようにする。

【 0 0 1 0 】

【作用】

動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値から最大値及び最小値を検出し、最大値及び最小値の和の 1 / 2 の値と参照プロツク及び候補プロツクの各画素の値とを比較演算して、参照プロツク及び候補プロツクの各画素の値を値「 1 」又は値「 0 」のコード値に符号化し、サーチエリア内で候補プロツクの位置をずらし、コード値を用いて参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を動きベクトルとして求め、直前に求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、最大最小値検出処理、符号化処理及び動きベクトル検出処理を実行して他の動きベクトルを算出する処理を n 回繰り返して n 個の動きベクトルを算出し、最後に求められた動きベクトルに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとし、これら複数の動きベクトルの和を最終的な動きベクトルとして出力するようにしたことにより、参照プロツク及び候補プロツクの各画素の値をコード値に変換することにより演算量を大幅に削減し得ると共に、順次サーチエリアを狭くしていくながら算出した複数の動きベクトルの和を最終的な動きベクトルとして出力することができるので、簡易な構成でかつ十分に高い精度の最終的な動きベクトルを算出し得る。

また、動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる全ての画素の値から最大値及び最小値を検出し、最大値及び最小値の差よりダイナミツクレンジを求め、参照プロツク及び候補プロツクの画素値と最大値又は最小値との差分を n ビットのコード値に符号化し、サーチエリア内で候補プロツクの位置をずらし、当該符号化されたコード値の最上位桁ビットを用いて参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を動きベクトルとして求め、直前に求められた動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられたコード値の桁よりも一つ下の桁のビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする処理を n 回 (n は 0 以上の整数) 繰り返し、これ

10

20

30

40

50

らによつて求められた動きベクトルのいずれかに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとし、これら複数の動きベクトルの和を最終的な動きベクトルとして出力するようにしたことにより、参照プロツクと全ての候補プロツクとに含まれる全ての画素の値から検出した最大値及び最小値の差よりダイナツクレンジを求め、参照プロツク及び候補プロツクの画素値と最大値又は最小値との差分を n ビットのコード値に符号化し、サーチエリアを次第に小さくしながら当該コード値の最上位桁のビットからその一つ下の桁のビットまで順番に用いて参照プロツク及び候補プロツク毎のマツチング演算を行うことにより演算量を大幅に削減し、順次サーチエリアを狭くしていきながら算出した全ての動きベクトルの和を最終的な動きベクトルとして出力することができる、簡易な構成でかつ十分に高い精度の最終的な動きベクトルを算出し得る。

【0011】

さらに、入力画像データに対してラプラシアンフィルタ処理が施された画像から、動きベクトルを検出するプロツクマツチングの対象となる参照プロツクとサーチエリア内の全ての候補プロツクとに含まれる画素の値の絶対値から最大値を検出し、最大値を参照プロツクとサーチエリア内の全ての候補プロツクのダイナミツクレンジとして、各参照プロツク及び各候補プロツクの画素値を、極性を表す符号を含む（符号 + n ）ビットのコード値に符号化し、サーチエリア内で候補プロツクの位置を順次ずらし、コード値の符号ビットを用いて参照プロツクと候補プロツクとのマツチング演算を行い、当該演算値が最小の位置を第1の動きベクトルとし、第1の動きベクトルで動き補償を行つた後、サーチエリアをより小さくして、コード値の最上位桁ビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を第2の動きベクトルとし、直前に求められた第2の動きベクトルで動き補償を行つた後、直前のサーチエリアより小さいサーチエリアで、直前に用いられたコード値の桁よりも一つ下の桁のビットを用いて、参照プロツク及び候補プロツク毎のマツチング演算を行い、当該演算値が最小の位置を他の動きベクトルとする処理を n 回（ n は 0 以上の整数）繰り返し、これらによつて求められた動きベクトルのいずれかに応じて動き補償を行つた後、参照プロツク及び候補プロツクそれぞれの画素値を用いて、最後に求められたサーチエリアに対して上下左右、所定画素数の範囲縮小したサーチエリアについてマツチング演算を行い、当該演算値が最小の位置を画素値による動きベクトルとし、これら複数の動きベクトルの和を最終的な動きベクトルとして出力するようにしたことにより、入力画像データに対してラプラシアンフィルタ処理を施してエッジ等の画像の特徴成分を強調したうえで動きベクトルを検出するプロツクマツチングの対象となる参照プロツクにおける画素値及び候補プロツクにおける画素値を、極性を表す符号を含む（符号 + n ）ビットのコード値に符号化し、当該（符号 + n ）ビットのコード値を用いて演算量を大幅に減らし、順次分解能を上げるようにして画像の特徴を考慮した一段と高精度の最終的な動きベクトルを得ることができる。

【0012】

【実施例】

以下図面について、本発明の一実施例を詳述する。

【0013】

（1）動きベクトル検出の原理

一般的に動きベクトルを検出する方法として、プロツクマツチングが行われる。これは、図1に示すように、現在のフレーム（又はフィールド） F_1 を m 画素 \times n 画素の大きさのプロツクに分割して得られる参照プロツク B_1 と、過去のフレーム（又はフィールド） F_2 のサーチエリア内（ $\pm s$ 画素）に存在する候補プロツク B_2 との画素毎の差分の絶対値和を評価値 P （ h, v ）として演算し、次式

【数1】

$$P(h, v) = \sum_{i=1}^m \sum_{j=1}^n |x(k)_{i,j} - x(k-1)_{i,j}| \dots \dots \quad (1)$$

に示すように、サーチエリア内で候補プロツク B 2 を 1 画素毎にずらしてトータル ($2s + 1$)² 点の評価値 $P(h, v)$ を演算する。これらの評価値 $P(h, v)$ の最小値を求めるこによつて、その最小値が示す相対的座標値 (h, v) を該参照プロツク B 1 の動きベクトルとする手法である。トータルの演算数は、演算量 = プロツク数 × サーチポイント数 × 評価式数で表される。この場合、サーチポイント数は $(2s + 1)^2$ であり、評価式数は減算 ($m \times n$)、絶対値演算 ($m \times n$)、加算 ($m \times n - 1$) の和となる。

【0014】

10

ここで動き検出のハードウェアを小さくするためには、上式の各項目を減らす必要があるが、プロツク数は変えることができないので、サーチポイント数又は評価式数を減らすしかない。サーチポイント数を減らす手法としては、3ステップ方式や正射影方式がある。またサーチポイント数と評価式数を同時に減らす手法として階層化方式がある。しかしながらマツチング演算自体は画素のレベル (8ビット) そのもので行われているため、さらにハードウェアを小さくするためこの実施例ではマツチング演算の対象となる画素の語長を減らすようになされている。

【0015】

(2) 第1実施例の動きベクトル検出方法及び装置

20

この第1実施例の動きベクトル検出方法では、時間的に連続する2枚のフレーム (又はフィールド) F 1、F 2 の画像から、現在のフレーム F 1 については m 画素 × n 画素の大きさのプロツクに分割し、その中のある参照プロツク B 1 に関して過去のフレーム (又はフィールド) F 2 からの動きベクトルを算出することを前提とする。現在のフレーム (又はフィールド) F 1 の参照プロツク B 1 のデータは、上述の分割したプロツクからあるプロツクを順次選択して供給し、過去のフレーム (又はフィールド) F 2 での候補プロツク B 2 のデータは参照プロツク B 1 の空間的位置と同じ位置を中心としてサーチエリア ($\pm s$ 画素) SA の中を順次動かして供給する。

【0016】

30

この第1実施例の動きベクトル検出方法では、次のような処理を行つて動きベクトルを算出する。まず参照プロツク B 1 の画素値とサーチエリア SA 内の全候補プロツク B 2 の画素値とにおける画素毎の差分の絶対値和を評価値 $P(h, v)$ として演算し、その評価値 $P(h, v)$ の最大値及び最小値を検出し、その最大値及び最小値の和の $1/2$ をしきい値として、各プロツクの画素値とそれぞれ比較演算することによつて、1ビット A D R C (adaptive dynamic range coding) 処理を行うことにより、参照プロツク B 1 及び候補プロツク B 2 の画素値を、値「1」又は値「0」のコード値に符号化する。

【0017】

40

次に各プロツクのコード値を用いて参照プロツク B 1 と候補プロツク B 2 の画素位置毎のマツチング演算として差分の絶対値和を求め、評価値 $P(h, v)$ を算出する。なお評価値の計算は、コード値の排他的論理和 (EXOR) をとつて求めて良い。この操作をサーチエリア SA 内で候補プロツク B 2 の位置をずらしながら、順次評価値を算出し、トータルで $(2s + 1)^2$ 点の評価値 $P(h, v)$ を演算する。次に求められた評価値に基づいて最小値の位置を検出し、その相対的な座標値 (h, v) を第1の動きベクトルとする。

【0018】

50

続いてこのようにして求められた第1の動きベクトルで動き補償を行つた後、決定された第1の候補プロツク B 2 を中心として、改めてサーチエリア SA をより小さく設定する。そして上述の操作を再度行うことによつて第2の動きベクトルを算出する。以下、適当な回数だけ、求めた動きベクトルで動き補償すると共にサーチエリア SA をより小さくして上述の操作を繰り返すことによつて、第 ($n - 1$) の動きベクトルを算出する。次に第 ($n - 1$) の動きベクトルを使用して、さらに動き補償を行つた後、最終段では、画素値そ

のものを用いてサーチエリア S A の中心を動かし、当該中心から上下左右方向に 1 画素単位で動かした範囲内で従来のプロツクマッチングを行い第 n の動きベクトルを算出する。最終的な動きベクトルは、上述で求めた第 1 ~ 第 n の動きベクトルの和を計算することで求められる。

【 0 0 1 9 】

ここで、図 2 にこの実施例の動きベクトル検出装置 1 の概略構成を示す。この動きベクトル検出装置 1 では、動きベクトル検出の前提として、入力される画像データについて走査変換回路 2 及びフレームメモリ 3 で、現在のフレーム（又はフィールド）F 1 と過去のフレーム（又はフィールド）F 2 のプロツクのデータを生成し、これを動きベクトル検出回路 4 に供給する。この動きベクトル検出回路 4 は、図 3 に示すように、大きく分けて 3 つの部分からなり、第 1 の動きベクトル $m v_1$ を求める動きベクトル算出回路 5、第 2 の動きベクトル $m v_2$ を求める動きベクトル算出回路 6、そして第 3 の動きベクトル $m v_3$ を算出する残りの回路から構成されている。

【 0 0 2 0 】

このうち動きベクトル算出回路 5 及び 6 については同様で、図 4 に示すように構成されている。この動きベクトル算出回路 5 及び 6 は、入力された現在の参照プロツク B 1 の画素値及び過去の候補プロツク B 2 の画素値に基づき最大最小値回路 7 で参照プロツク B 1 の画素値とサーチエリア S A 内の全候補プロツク B 2 の画素値とにおける画素毎の差分の絶対値和を評価値 $P(h, v)$ として演算し、その評価値 $P(h, v)$ の最大値及び最小値を検出し、それぞれレジスタ 8、9 に保持する。これによりレジスタ 8、9 に保持された最大値及び最小値は、加算器 10 で加算された後ビットシフトされて 1 / 2 倍され、その結果をレジスタ 11 に保持する。レジスタ 11 の値は、1 ビット A D R C を行うためのしきい値となる。

【 0 0 2 1 】

このしきい値が算出されるまでの遅延分を F I F O 1 2、1 3 で補償し、比較回路 1 4、1 5 によって現在及び過去プロツクデータとしきい値が比較され、しきい値より大きいときは値「1」、小さいときは値「0」のコードデータが出力される。これにより 1 ビット A D R C が実行される。比較回路 1 4、1 5 の出力は、一旦メモリ回路 1 6、1 7 に記憶される。

【 0 0 2 2 】

メモリ回路 1 6、1 7 の読み出しへは、現在プロツクのコードデータとサーチエリアの範囲内で切り出した過去プロツクのコードデータとが順次出力され、評価値演算回路 1 8 で画素位置毎の演算値の積算が行われる。評価値演算回路 1 8 ではイクスクルーシブオア（E X O R）回路 1 9 で、排他的論理和によるマッチングの度合いが計られる。つまりコードデータが一致していると値「0」が出力され、一致していないと値「1」が出力され積算される。この評価値演算回路 1 8 と同じ動作をするものが評価値演算回路 2 0（図 3）で、ここでは差分の絶対値を積算する。

【 0 0 2 3 】

1 プロツクのコードデータが走査された後、レジスタ 2 2 にあるサーチポイントでの評価値が保持されていることになる。この評価値は評価値メモリ 2 3 に記憶される。このような演算をサーチポイントをずらしながら行つていくと、評価値メモリ 2 3 には、全部で $(2s + 1)^2$ 点の評価値 $P(h, v)$ が記憶される。次に評価値メモリ 2 3 から評価値を読み出して、最小値検出回路 2 4 で最小値の位置を検出する。このときの相対的な座標が求められ、ベクトル決定回路 2 5 で第 1 又は第 2 の動きベクトル $m v_i$ が出力される。

【 0 0 2 4 】

一方動きベクトル算出回路 5 で第 1 の動きベクトル $m v_1$ が算出されるまでの時間を F I F O 2 6 及びメモリ回路 2 7 で補償すると共に、メモリ回路 2 7 の読み出しへは、第 1 の動きベクトル $m v_1$ で動き補償してサーチエリアの中心を動かし、さらにサーチエリアの範囲を縮小した形でアドレスコントロール 2 8 でアドレスを指定して、プロツクのデー

10

20

30

40

50

タを順次動きベクトル算出回路 6 に入力する。この結果動きベクトル算出回路 6 では、上述の動きベクトル算出回路 5 と同様にして第 2 の動きベクトル $m v_2$ が算出される。

【 0 0 2 5 】

第 2 の動きベクトル $m v_2$ が算出されると、同様に FIFO 29 及びメモリ回路 30 で遅延を補償すると共に、メモリ回路 30 の読み出しは、第 2 の動きベクトル $m v_2$ で動き補償してサーチエリアの中心を動かし、当該中心から上下左右方向に 1 画素単位で動かした範囲内で、アドレスコントロール 31 でアドレスを指定してプロックのデータを順次メモリ回路 32 及び 33 に出力する。

【 0 0 2 6 】

この最終段の動きベクトル検出は、第 1 及び第 2 の動きベクトル検出とは異なり、例えば 10 8 ビットでなる画素値そのもので行う。メモリ回路 32 及び 33 に一旦記憶された現在プロックデータとサーチエリアの範囲内で切り出した過去プロックデータは順次出力され、評価値演算回路 20 の減算回路 34 及び絶対値回路 35 で画素位置毎の差分の絶対値の演算が行われ、その結果が加算器 36 及びレジスタ 37 で、順次画素毎の積算が行われる。

【 0 0 2 7 】

1 プロックのコードデータが走査された後、レジスタ 37 にあるサーチポイントでの評価値が保持されたことになる。この評価値は評価値メモリ 38 に記憶される。このような演算をサーチポイントをずらしながら行つていくと、評価値メモリ 38 には、例えば 9 点の評価値が記憶される。そして評価値メモリ 38 から評価値を読み出して、最小値検出回路 39 で評価値が最小となる位置を検出する。このときの相対的な座標が求められ、ベクトル決定回路 40 で第 3 の動きベクトル $m v_3$ が出力される。第 1 の動きベクトル $m v_1$ 、第 2 の動きベクトル $m v_2$ 及び第 3 の動きベクトル $m v_3$ は加算器 41、42 で加算され、当該加算結果が入力画像データの動きベクトル $m v$ として、レジスタ 43 に保持され外部に出力される。 20

【 0 0 2 8 】

以上の構成によれば、動きベクトルを検出するプロックマッチングの対象となる参照プロック B 1 と候補プロック B 2 を 1 ビット A D R C で符号化し、そのコード値によつてマッチング演算を行つて動きベクトル $m v_1$ を算出し、その動きベクトル $m v_1$ で動き補償した後、サーチエリアをより小さくして、さらにプロックを小さくして繰り返して動きベクトル $m v_2$ を求め、最終的には繰り返し求めた動きベクトル $m v_2$ に応じて動き補償した後、参照プロック B 1 及び候補プロック B 2 それぞれの画素値を用いて、サーチエリアの ± 1 画素の範囲でマッチング演算を行つて動きベクトル $m v_3$ を求め、全ての動きベクトル $m v_1$ 、 $m v_2$ 、 $m v_3$ の和を出力すべき動きベクトル $m v$ として求めることにより、簡易な構成でかつ十分に高い精度で動きベクトル $m v$ を算出し得る。 30

【 0 0 2 9 】

(3) 第 2 実施例の動きベクトル検出方法及び装置

この第 2 実施例の動きベクトル検出方法は、現在及び過去プロックのデータについて第 1 実施例の 1 ビット A D R C による符号化に代えて、n ビット A D R C のコード値に符号化するものである。実際上次のような操作を行つて動きベクトルを算出する。まず参照プロック B 1 とサーチエリア内の全候補プロック B 2 のデータから最大値及び最小値を検出し、その最大値及び最小値の差からダイナミックレンジを求める。このダイナミックレンジを 2^n で割り算して量子化ステップ幅を求め、各プロックの画素値と最小値又は最大値との差分データをその量子化ステップ幅で割り算し、n ビットのコードに符号化して n ビット A D R C 符号化を行う。なお A D R C 符号化は R O M と簡単なロジックで実現される。 40

【 0 0 3 0 】

このようにして n ビット A D R C 符号化された参照プロック B 1 と候補プロック B 2 との A D R C コード値の M S B (最上位桁ビット) で、画素位置毎のマッチング演算として排他論理和 (E X O R) 演算し、その結果のプロック内積算値として評価値 P (h, v) を 50

算出する。この操作をサーチエリア内で候補プロツク B 2 の位置をずらしながら順次評価値を算出し、トータルで $(2s + 1)^2$ 点の評価値 $P(h, v)$ を演算する。次に求められた評価値から最小値の位置を検出し、その相対的な座標値 (h, v) を第 1 の動きベクトルとする。

【0031】

続いてこのようにして求められた第 1 の動きベクトルで動き補償を行つた後、決定された第 1 の候補プロツクを中心として、改めてサーチエリアをより小さく設定する。そして A D R C コード値の最上位桁より 2 番目のビットで同様にして、プロツク毎のマッチング演算を行い、順次サーチエリア内で評価値 $P(h, v)$ を算出し、その評価値 $P(h, v)$ の最小点を検出することで第 2 の動きベクトルを算出する。

10

【0032】

以下、順次 A D R C コード値の順次下位桁のビットについてのマッチング演算を、サーチエリアをより小さくして繰り返すことによつて、第 n の動きベクトルまで算出する。この第 n の動きベクトルを使用して、さらに動き補償を行つた後、最終段では、画素値そのものを用いてサーチエリアの ± 1 画素の範囲で従来と同様のプロツクマッチングを行い、第 $(n + 1)$ の動きベクトルを算出する。最終的な動きベクトルは、第 1 ~ 第 $(n + 1)$ の動きベクトルの和を計算することで求めるようになされている。

【0033】

ここでこの実施例の動きベクトル検出装置 1 は図 2 と同様の構成でなり、動きベクトル検出回路 4 が、図 5 及び図 6 に示すように、大きく分けて 4 つの部分からなり、A D R C 符号化を行う A D R C 符号化回路 5 0、第 1 の動きベクトルを算出する第 1 の動きベクトル算出回路 5 1 及び第 2 の動きベクトルを算出する第 2 の動きベクトル算出回路 5 2、そして第 3 の動きベクトルを算出する残りの回路から構成されている。

20

【0034】

まず A D R C 符号化回路 5 0 において、入力された現在の参照プロツク B 1 の画素値及びサーチエリア内で切り出された過去の候補プロツク B 2 の画素値に基づき最大最小値回路 5 3 で参照プロツク B 1 の画素値とサーチエリア内の全候補プロツク B 2 の画素値とにおける画素毎の差分の絶対値和を評価値 $P(h, v)$ として演算し、その評価値 $P(h, v)$ の最大値及び最小値を検出し、それぞれレジスタ 5 4、5 5 に保持する。保持された最大値及び最小値を減算器 5 6 で減算してダイナミツクレンジを求めた後、そのダイナミツクレンジをレジスタ 5 7 に保持する。

30

【0035】

現在プロツクデータに関しては遅延メモリ 5 8 によつて適当な遅延の後、減算器 6 0 において最小値を減算されレジスタ 6 2 に保持する。同様に過去プロツクデータに関しては遅延メモリ 5 9 によつて遅延された後、減算器 6 1 において最小値を減算されレジスタ 6 3 に保持する。各々のデータは、それぞれダイナミツクレンジデータと共に A D R C 変換用 R O M 6 4 及び 6 5 に入力されて 2 ビット A D R C 符号化され、それぞれレジスタ 6 6 及び 6 7 にコード値が保持される。

【0036】

保持されたコード値は、その M S B (最上位桁ビット) が第 1 の動きベクトル算出回路 5 1 に、そして最上位桁より 2 ビット目 (2 n d M S B、この場合は L S B となる) が遅延メモリ 6 8、6 9 を経て第 2 の動きベクトル算出回路 5 2 に供給される。第 1 及び第 2 の動きベクトル算出回路 5 1 及び 5 2 は、同様の回路で構成されている。例えば第 1 の動きベクトル算出回路 5 1 において、入力された現在の A D R C コード値の M S B と過去の A D R C コード値の M S B が E X O R ゲート 7 0 で比較され、一致した場合は値「0」の出力、異なる場合は値「1」の出力が次段の加算器 7 1 及びレジスタ 7 2 でプロツク内のデータ数の回数だけ積算される。なお、加算器 7 1 及びレジスタ 7 2 に代えてカウンタを用いても良い。

40

【0037】

1 プロツクのコードデータが走査された後、レジスタ 7 2 にはあるサーチポイントでの評

50

価値が保持されていることになる。レジスタ72の出力は一旦評価値テーブルメモリ73に記憶される。以上の演算をサーチポイントをずらしながら行つていくと、評価値テーブルメモリ73には、全部で $(2s+1)^2$ 点の評価値P(h, v)が記憶されていることになる。次に評価値メモリ回路73から評価値P(h, v)を読み出して、ベクトル決定回路74で評価値が最小となる位置を検出する。これにより最小値の位置の相対的な座標が求められ、第1の動きベクトルmv₁が outputされる。

【0038】

一方、第1の動きベクトルmv₁が算出されるまでの時間を遅延メモリ68、69で補償すると共に、遅延メモリ69の読み出しは、第1の動きベクトルmv₁を考慮してサーチエリアの中心を動かし、さらにサーチ範囲を縮小した形でアドレスコントロール75でアドレスを指定してデータを出力する。これにより第2の動きベクトル検出回路52にデータが入力され、動きベクトル検出回路51と同様にして第2の動きベクトルmv₂が算出される。

【0039】

第2の動きベクトルmv₂が算出されると、遅延メモリ76及び77からの画素データは同様に遅延が補償されると共に、遅延メモリの読み出しは、加算器78及びレジスタ79を通じて得られる第1及び第2の動きベクトルmv₁及びmv₂の和を考慮してサーチエリアの中心を動かし、当該中心から上下左右方向に1画素単位で動かした範囲内で、アドレスコントロール80でアドレスを指定して行われる。

【0040】

最終段の動きベクトル検出は、第1及び第2の動きベクトルmv₁及びmv₂の検出とは異なり、例えば8ビットでなる画素値そのもので行われる。すなわち遅延メモリ76、77に一旦記憶された現在プロツクデータとサーチエリアの範囲内で切り出した過去プロツクデータは順次出力され、減算器81で画素位置毎の差分が計算され、絶対値化回路82で絶対値が演算される。その結果が加算器83を通してレジスタ84に保持されて、順次画素毎の積算が行われる。1プロツクのコードのコードデータが走査された後、レジスタ84にはあるサーチポイントでの評価値が保持されることになる。

【0041】

この評価値は評価値テーブルメモリ85に記憶される。このような演算をサーチポイントをずらしながら行つていくと、評価値メモリ85には例えば9点の評価値が記憶される。次に評価値テーブルメモリ85から評価値を読み出して、ベクトル決定回路86で最小値の位置を検出し、このときの相対的な座標が求められ第3の動きベクトルmv₃が outputされる。第1の動きベクトルmv₁、第2の動きベクトルmv₂及び第3の動きベクトルmv₃は加算器87で加算され、当該加算結果が入力画像データの動きベクトルmvとしてレジスタ88に保持され、外部に出力される。

【0042】

以上の構成によれば、動きベクトルを検出するプロツクマッチングの対象となる参照プロツクB1と候補プロツクB2をnビットADRCで符号化し、そのコード値のMSBによってマッチング演算を行つて動きベクトルmv₁を算出し、その動きベクトルmv₁で動き補償した後、サーチエリアをより小さくし、さらにプロツクを小さくして繰り返し最上位桁より2ビット目のビットを用いてマッチング演算して動きベクトルmv₂を求め、最終的には繰り返し求めた動きベクトルに応じて動き補償した後、参照プロツクB1及び候補プロツクB2それぞれの画素値を用い、サーチエリアの中心を動かし、当該中心から上下左右方向に1画素単位で動かした範囲内でマッチング演算を行つて動きベクトルmv₃を求め、全ての動きベクトルmv₁、mv₂、mv₃の和を出力すべき動きベクトルmvとして求めるようにしたことにより、簡易な構成でかつ十分に高い精度で動きベクトルmvを算出し得る。

【0043】

(4) 第3実施例の動きベクトル検出方法及び装置

この第3実施例の動きベクトル検出方法においては、入力画像に対してラプラシアンフィルタ

10

20

30

40

50

ルタ処理を施した後、動きベクトルを検出する。実際上ラプラシアンフィルタ処理の後、参照プロツクとサーチエリア内の全候補プロツクのデータは、0 レベルを中心として + の値と - の値に変化しており、特に画像のエッジ付近での変化が大きい。そこで、この実施例では、A D R C による符号化をする際に、極性を考慮した符号化を行う。

【 0 0 4 4 】

先ず、参照プロツクとサーチエリア内の全候補プロツクのデータの絶対値の最大値を検出し、当該最大値をダイナミックレンジとする。このダイナミックレンジを 2^n で割り算して量子化ステップ幅を求め、各プロツク内の画素値を当該量子化ステップ幅で割り算することにより (符号 + n) ビットのコードに符号化する。すなわち (符号 + n) ビットの A D R C 符号化処理を行う。この場合、符号ビットは、正のときに「0」とし、負のときに「1」とする。なお、この (符号 + n) ビットの A D R C 符号化は R O M と簡単なロジックで実現できる。

【 0 0 4 5 】

このように (符号 + n) ビット A D R C 符号化された参照プロツクと候補プロツクの A D R C コード値のうち符号ビットを用いて、画素位置毎のマッチング演算として排他的論理和 (E X O R) 演算を行い、その結果のプロツク内積算値として評価値 P (h, v) を算出する。この処理をサーチエリア内で候補プロツクの位置をずらしながら行うことにより順次評価値を算出し、トータルで $(2s+1)^2$ 点の評価値 P (h, v) を算出する。次に、求められた評価値が最小となる位置を検出し、その相対的な座標値 (h, v) を第 1 の動きベクトルとする。

【 0 0 4 6 】

続いてこのようにして求められた第 1 の動きベクトルで動き補償を行つた後、サーチエリアをより小さく設定する。そして A D R C コード値の M S B (最上位桁ビット) で同様にして画素位置毎のマッチング演算 (排他的論理和演算) を行い、順次サーチエリア内で評価値を算出し、当該評価値の最小点を検出することで第 2 の動きベクトルを算出する。

【 0 0 4 7 】

以下、順次 A D R C コード値の下位桁についてのマッチング演算を、サーチエリアをより小さくして繰り返すことによつて、第 (n + 1) の動きベクトルまで算出する。次にこの第 (n + 1) の動きベクトルを使用して、さらに動き補償を行つた後、最終段では、画素値そのものを用いてサーチエリアの ± 1 画素の範囲で従来と同様のプロツクマッチングを行い、第 (n + 1) の動きベクトルを算出する。そして最終的な動きベクトルは、第 1 ~ 第 (n + 1) の動きベクトルの和を計算することで求めるようになされている。

【 0 0 4 8 】

ここでこの実施例の動きベクトル検出装置 100 は、図 7 に示すように構成されており、入力画像データをラプラシアンフィルタ 101 に入力し、入力画像データに対してラプラシアンフィルタ 101 によってラプラシアンフィルタ処理を施してから動きベクトルを検出する。動きベクトル検出装置 100 はラプラシアンフィルタ処理後の画像データに対して走査変換回路 102 及びフレームメモリ 103 で、現在のフレーム (又はフィールド) F 1 と過去のフレーム (又はフィールド) F 2 のプロツクのデータを形成し、これらを動き検出回路 104 に送出する。

【 0 0 4 9 】

ここでラプラシアンフィルタ 101 は、例えば図 8 に示すような 2 次元のフィルタ係数とされている。実際上図 8 (A) の 2 次元フィルタ係数を実現するためには、ラプラシアンフィルタ 101 を、図 9 に示すように構成すれば良い。すなわちラプラシアンフィルタ 101 は、入力画像データをそれぞれ 1 画素分の遅延時間を有する遅延素子 (D) 105 ~ 107 を介して 1 ライン分の遅延時間を有するラインディレイ (L i n e) 112 に送出する。そして各遅延素子 105 ~ 107 の出力をそれぞれ乗算係数が - 1 に選定された乗算回路 108 ~ 110 を介して積算回路 111 に送出する。

【 0 0 5 0 】

ラインディレイ 112 の出力は、それぞれ 1 画素分の遅延時間を有する遅延素子 113 ~ 50

115を介して1ライン分の遅延時間を有するラインディレイ119に送出される。そして各遅延素子113～115の出力が、それぞれ乗算係数が-1、8、1に選定された乗算回路116、117、118を介して積算回路111に送出される。さらにラインディレイ119の出力は、それぞれ1画素分の遅延時間を有する遅延素子120～122に順次送出される。そして各遅延素子120～122の出力が、それぞれ乗算係数が-1に選定された乗算回路123～125を介して積算回路111に送出される。この結果積算回路111において積算された各乗算回路108～110、116～118及び123～125の出力がラプラスアンフィルタ101の出力として遅延素子126を介して出力される。

【0051】

10

またこの実施例の動きベクトル検出回路104は、図10及び図11に示すように、大きく分けて4つの部分からなり、(符号+n)ビットのADC符号化処理を行うADC符号化回路130、第1の動きベクトル $m v_1$ を求める第1の動きベクトル算出回路131、第2の動きベクトルを求める第2の動きベクトル算出回路132、第3の動きベクトル $m v_3$ を求める第3の動きベクトル算出回路133、そして第4の動きベクトル $m v_4$ を求める残りの回路から構成されている。

【0052】

まずADC符号化回路130について説明する。ADC符号化回路130は入力された現在のプロツク化された画素データ及びサーチエリア内で切り出された過去のプロツク化された画素データを絶対値化回路134でそれぞれ絶対値化した後、続く最大値検出回路135で最大値を検出し、これをダイナミックレンジとしてレジスタ136に保持する。また現在プロツクデータに関しては遅延回路137によって適当に遅延した後順次レジスタ138に保持する。同様に過去のプロツクデータに関しては遅延回路139によって遅延した後レジスタ140に保持する。このレジスタ138及び140に保持された各々のデータはそれぞれダイナミックレンジデータと共にADC変換用ROM141及びADC変換用ROM142に入力され、(符号+2)ビットADC符号化され、それぞれレジスタ143及び144にコード値が保持される。

20

【0053】

保持されたコード値は、その符号ビットがそのまま第1の動きベクトル算出回路131に、MSBが遅延メモリ145及び146を介して第2の動きベクトル算出回路132に、そして2ndMSB(この場合はLSBとなる)が遅延メモリ147及び148を介して第3の動きベクトル算出回路133に供給される。ここで第1、第2及び第3の動きベクトル算出回路131、132及び133は、第2実施例において上述した第1及び第2の動きベクトル算出回路51及び52(図6)と同様の構成であり、入力されたADCコード値に対して排他的論理和(EXOR)演算を行い、その結果のプロツク内積算値として評価値P(h,v)を算出し、この評価値が最小となる位置を検出し、その相対的な座標値(h,v)をそれぞれの動きベクトルとする。

30

【0054】

動きベクトル検出回路104においては、第1の動きベクトル $m v_1$ 、第2の動きベクトル $m v_2$ 、第3の動きベクトル $m v_3$ 、第4の動きベクトル $m v_4$ の順に順次動き補償をしながらサーチエリアを小さくして動きベクトルを求ることにより最終的な動きベクトル $m v$ を求めるのに要する演算量を低減するようになされている。

40

すなわち動きベクトル検出回路104は、第1の動きベクトル算出回路131によって第1の動きベクトル $m v_1$ が求められると、これを加算器149及びアドレスコントロール150に送出する。そしてアドレスコントロール150は第1の動きベクトル $m v_1$ を考慮して、遅延メモリ146にサーチエリアの中心を動かすと共にサーチエリアを縮小した形のアドレスを指定して遅延メモリ146からデータを出力させる。

【0055】

また第2の動きベクトル $m v_2$ が算出されると、これが加算器149に送出され、ここで第1及び第2の動きベクトル $m v_1$ 及び $m v_2$ の和が算出され、これがレジスタ1

50

51を介して加算器152及びアドレスコントロール153に送出される。アドレスコントロール153は第1及び第2の動きベクトル $m v_1$ 及び $m v_2$ の和を考慮して、遅延メモリ148にサーチエリアの中心を動かすと共にサーチエリアを縮小した形のアドレスを指定して遅延メモリ148からデータを出力させる。

【0056】

さらに第3の動きベクトル $m v_3$ が算出されると、これが加算器152に送出され、ここで第1、第2及び第3の動きベクトル $m v_1$ 、 $m v_2$ 及び $m v_3$ の和が算出され、これがレジスタ154を介して加算器155及びアドレスコントロール156に送出される。ここで遅延回路137及び139からの画素データは遅延メモリ157及び158によって上述の第1、第2及び第3の動きベクトル $m v_1$ 、 $m v_2$ 及び $m v_3$ を算出する時間分の遅延が補償されて格納されている。そしてアドレスコントロール156が、第1、第2及び第3の動きベクトル $m v_1$ 、 $m v_2$ 及び $m v_3$ の和を考慮して、遅延メモリ158にサーチエリアの中心を動かすと共にサーチエリアを縮小した形のアドレスを指定して遅延メモリ156からデータを出力させる。
10

【0057】

最終段での動きベクトル検出（すなわち第4の動きベクトル $m v_4$ の検出）は、第1、第2及び第3の動きベクトルの検出とは異なり、（符号+8）ビットの画素値そのもので行う。すなわち遅延メモリ157及び158に一旦格納された現在プロツクデータとサーチエリア内で切り出された過去のプロツクデータが順次出力され、減算器159で画素位置毎の差分が計算され、絶対値化回路160で絶対値が演算される。その結果が加算器161を通してレジスタ162に保持されて、順次画素毎の積算が行われる。1プロツクのコードデータが走査された後、レジスタ162にはあるサーチポイントでの評価値が保持されていることになる。この評価値は評価値テーブルメモリ163に記憶される。このような演算をサーチポイントをずらしながら行つていくと、評価値テーブルメモリ163には例えれば9点の評価値が記憶される。
20

【0058】

次に評価値テーブルメモリ163から評価値を読み出し、ベクトル決定回路164によつて評価値が最小となる位置を検出し、このときの相対的な座標が求められ、第4の動きベクトル $m v_4$ が出力される。最終的に、第1、第2、第3及び第4の動きベクトル $m v_1$ 、 $m v_2$ 、 $m v_3$ 及び $m v_4$ は加算器155で加算され、この加算結果が入力画像データの動きベクトル $m v$ としてレジスタ165に保持され、外部に出力される。
30

【0059】

以上の構成において、この実施例の動きベクトル検出装置100は、入力画像データに対してラプラシアンフィルタ処理を施してエッジ等の画像の特徴成分を強調したうえで、動きベクトルを検出するプロツクマッチングの対象となる参照プロツクB1と候補プロツクB2を（符号+n）ビットADRCで符号化する。

【0060】

次に、そのADRCコード値のうち符号ビット（極性を表すビット）を用いて第1の動きベクトル $m v_1$ を算出し、その動きベクトル $m v_1$ で動き補償した後サーチエリアを小さくしてADRCコード値のうち最上位桁ビットを用いて第2の動きベクトル $m v_2$ を算出する。
40

【0061】

次に、第1及び第2の動きベクトル $m v_1$ 及び $m v_2$ の和に応じて動き補償した後サーチエリアを小さくしてADRCコード値のうち2ビット目のビットを用いて第3の動きベクトル $m v_3$ を算出する。次に第1、第2及び第3の動きベクトル $m v_1$ 、 $m v_2$ 及び $m v_3$ の和に応じて動き補償した後、参照プロツクB1及び候補プロツクB2それぞれの画素を用いて、サーチエリアの中心を動かし、当該中心から上下左右方向に1画素単位で動かした範囲内でマッチング演算を行つて第4の動きベクトル $m v_4$ を算出する。そして最後に、全ての動きベクトル $m v_1$ 、 $m v_2$ 、 $m v_3$ 、 $m v_4$ の和を出力すべき動きベクトル $m v$ とする。
50

【0062】

以上の構成によれば、ラプラシアンフィルタ処理を施した入力画像データについて、順次分解能を上げるようにして動きベクトル $m v$ を求めるようにしたことにより、画像の特徴を考慮した一段と高精度の動きベクトル $m v$ を得ることができる。

【0063】

(5) 他の実施例

なお上述の第1及び第2の実施例においては、第1及び第2の動きベクトルを1ビットA D R C又は n ビットA D R Cで符号化したコードデータでそれぞれ求め、第3の動きベクトルについて画素値そのものを用いて求めた場合について述べたが、さらに複数階層の動きベクトルについて1ビットA D R C又は n ビットA D R Cで符号化したコードデータでそれぞれ求め、最終的な動きベクトルのみを画素値そのものを用いて求めるようにしても良い。

10

【0064】

同様に、上述の第3実施例においては、(符号+2)ビットのコード値にA D R C符号化されたデータを用いたため、第3の動きベクトルまでをコード値を用いて求め、第4の動きベクトルを画素値そのものを用いて求めたが、さらに一般化して(符号+n)ビットのコード値にA D R C符号化されたデータを用いた場合には、第1の動きベクトルを符号ビットにより求め、第2~第(n+1)番目の動きベクトルを符号ビット以外のコード値を用いて求め、最終的な動きベクトルのみを画素値そのものを用いて求めるようにすれば良い。

20

【0065】

【発明の効果】

上述のように本発明によれば、参照プロツク及び候補プロツクの各画素の値をコード値に変換することにより演算量を大幅に削減し得ると共に、順次サーチエリアを狭くしていくながら算出した複数の動きベクトルの和を最終的な動きベクトルとして出力することができる、簡易な構成でかつ十分に高い精度の最終的な動きベクトルを算出し得る。

また本発明によれば、参照プロツクと全ての候補プロツクとに含まれる全ての画素の値から検出した最大値及び最小値の差よりダイナミックレンジを求め、参照プロツク及び候補プロツクの画素値と最大値又は最小値との差分を n ビットのコード値に符号化し、サーチエリアを次第に小さくしながら当該コード値の最上位桁のビットからその一つ下の桁のビットまで順番に用いて参照プロツク及び候補プロツク毎のマッチング演算を行うことにより演算量を大幅に削減し、順次サーチエリアを狭くしていくながら算出した全ての動きベクトルの和を最終的な動きベクトルとして出力することができる、簡易な構成でかつ十分に高い精度の最終的な動きベクトルを算出し得る。

30

【0066】

さらに本発明によれば、入力画像データに対してラプラシアンフィルタ処理を施してエッジ等の画像の特徴成分を強調したうえで動きベクトルを検出するプロツクマツチングの対象となる参照プロツクにおける画素値及び候補プロツクにおける画素値を、極性を表す符号を含む(符号+n)ビットのコード値に符号化し、当該(符号+n)ビットのコード値を用いて演算量を大幅に減らし、順次分解能を上げるようにして画像の特徴を考慮した一段と高精度の最終的な動きベクトルを得ることができる。

40

【図面の簡単な説明】

【図1】プロツクマツチングによる動きベクトル検出の原理の説明に供する略線図である。

【図2】本発明による動きベクトル検出装置の概略構成を示すプロツク図である。

【図3】第1実施例による動きベクトル検出回路を示すプロツク図である。

【図4】第1実施例による動きベクトル検出回路を示すプロツク図である。

【図5】第2実施例による動きベクトル検出回路を示すプロツク図である。

【図6】第2実施例による動きベクトル検出回路を示すプロツク図である。

【図7】第3実施例による動きベクトル検出装置の概略構成を示すプロツク図である。

50

【図 8】第3実施例で用いるラプラシアンフィルタのフィルタ係数の例を示す略線図である。

【図 9】ラプラシアンフィルタの構成例を示すプロツク図である。

【図 10】第3実施例による動きベクトル検出回路を示すプロツク図である。

【図 11】第3実施例による動きベクトル検出回路を示すプロツク図である。

【符号の説明】

1、100 ……動きベクトル検出装置、2、102 ……走査変換回路、3、103 ……フレームメモリ、4、104 ……動きベクトル検出回路、5、6、51、52、131、132、133 ……動きベクトル算出回路、7、53 ……最大最小値検出回路、8、9、11、22、37、43、54、55、62、63、66、67、72、72、79、84、88、136、138、140、143、144、151、154、162、165 ……レジスタ、10、21、41、36、42、71、71、78、83、87、149、152、155、161 ……加算器、12、13、26 ……FIFO、14、15 ……比較回路、16、17、27、30、32、33 ……メモリ、18、20 ……評価値演算回路、19、70、70 ……EXORゲート、23、38 ……評価値メモリ、24、39 ……最小値検出回路、25、40、74、74、86、164 ……ベクトル決定回路、28、31、75、80、150、153、156 ……アドレスコントロール、34、56、60、61、81、159 ……減算器、35、82、134 ……絶対値化回路、50、130 ……ADC符号化回路、58、59、68、69、76、77、137、139、145、146、147、148、157、158 ……遅延メモリ、64、65、141、142 ……ADCコード変換ROM、73、73、85、163 ……評価値テーブルメモリ、101 ……ラプラシアンフィルタ、105～107、113～115、120～122、126 ……デイレイ、108～110、116～118、123～125 ……乗算回路、111 ……積算回路、112、119 ……ラインデイレイ、135 ……最大値検出回路、 $m v_1$ ……第1の動きベクトル、 $m v_2$ ……第2の動きベクトル、 $m v_3$ ……第3の動きベクトル、 $m v_4$ ……第4の動きベクトル、 $m v$ ……動きベクトル。

【図1】

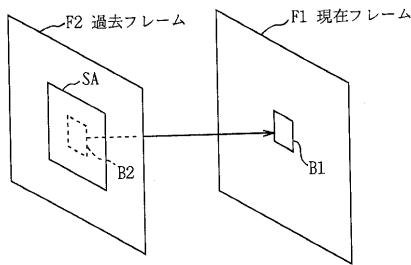


図1 ブロックマッチングによる動きベクトル検出

【図3】

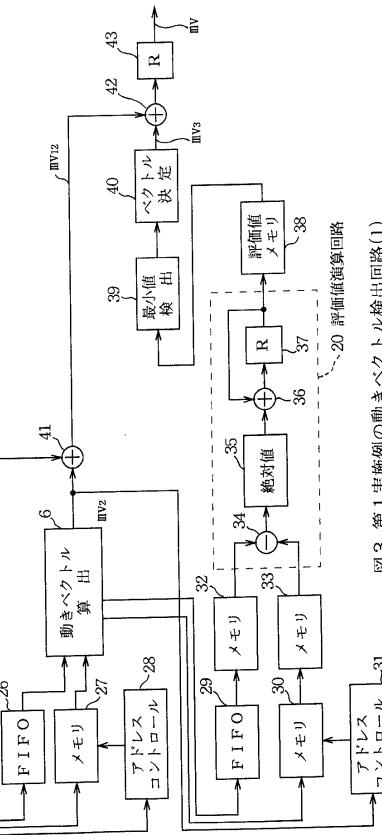


図3 第1実施例の動きベクトル検出回路(1)

【図2】

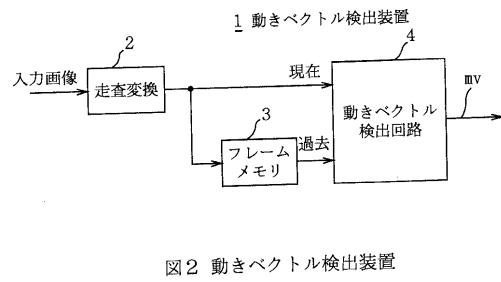


図2 動きベクトル検出装置

【図4】

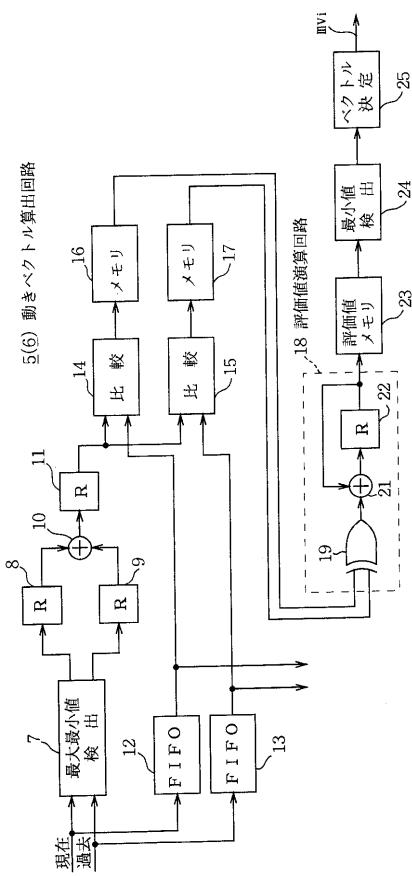


図4 第1実施例の動きベクトル検出回路(2)

【図5】

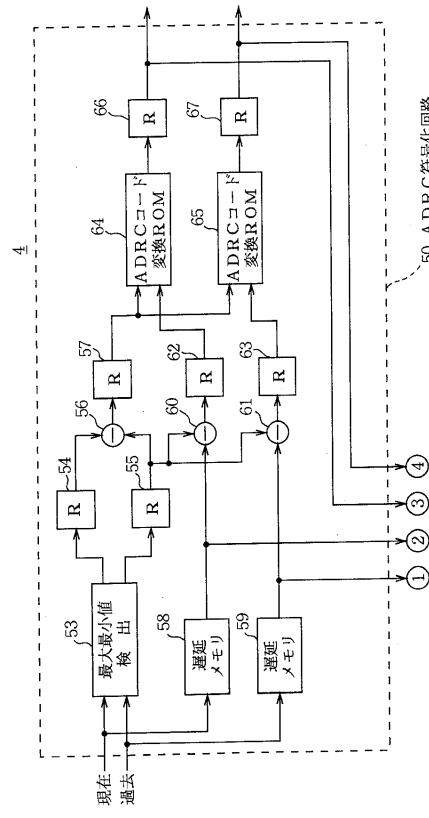


図5 第2実施例の動きベクトル検出回路(1)

【図 6】

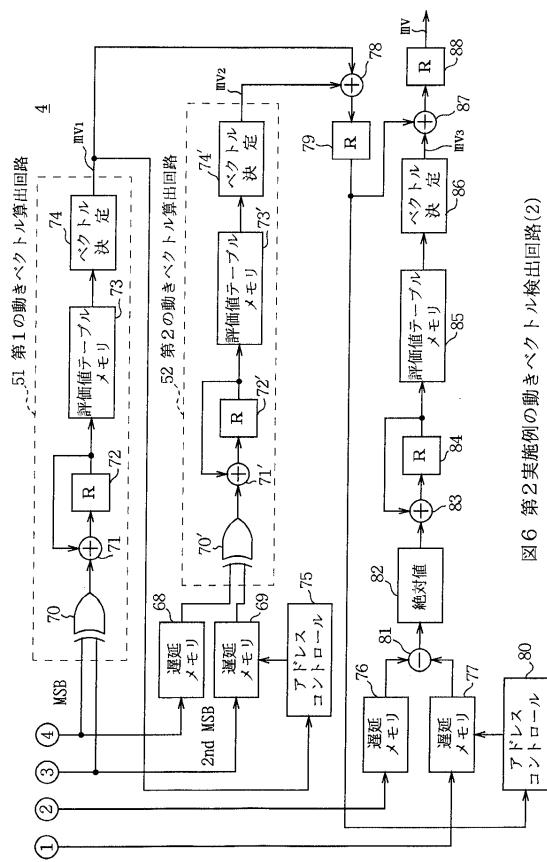


図6 第2実施例の動きベクトル検出回路(2)

【図 7】

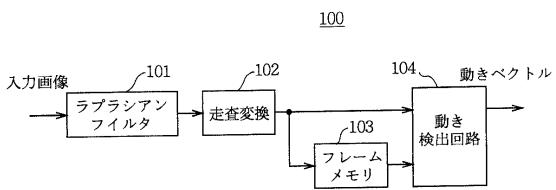


図7 第3実施例の動きベクトル検出装置

【図 8】

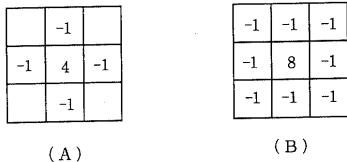


図8 2次元ラプラスアンフィルタの例

【図 9】

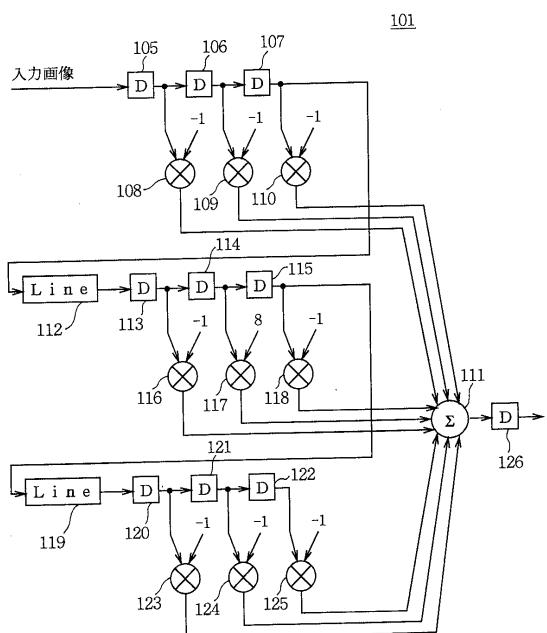


図9 ラプラスアンフィルタの構成例

【図 10】

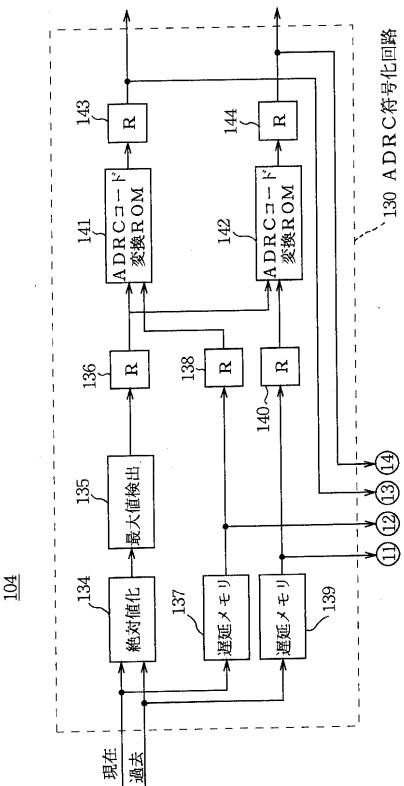


図10 第3実施例の動きベクトル検出回路 (1)

【 図 1 1 】

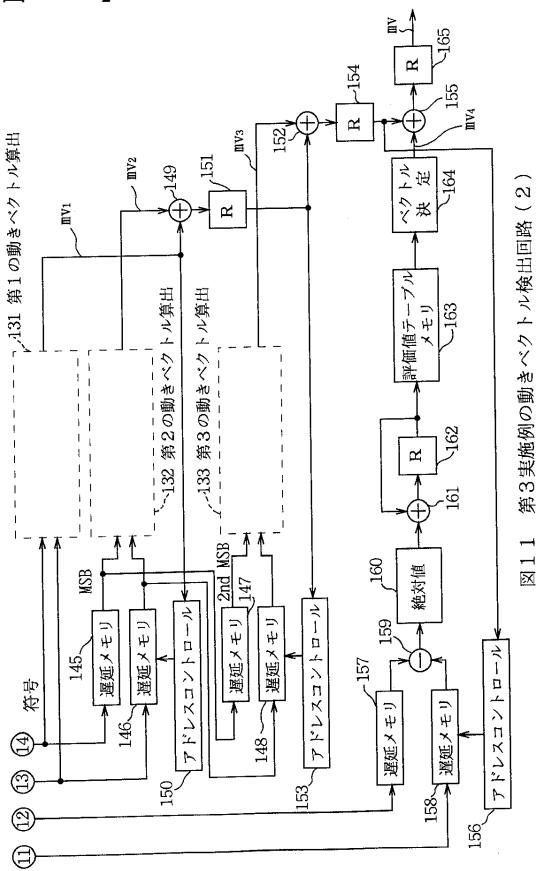


図11 第3実施例の動きベクトル検出回路(2)

フロントページの続き

(58)調査した分野(Int.Cl.⁷, DB名)

H04N 7/26-7/68