

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4004653号
(P4004653)

(45) 発行日 平成19年11月7日(2007.11.7)

(24) 登録日 平成19年8月31日(2007.8.31)

(51) Int. Cl. F I
H04N 7/32 (2006.01) H04N 7/137 Z

請求項の数 10 (全 17 頁)

(21) 出願番号	特願平10-218424	(73) 特許権者	595106811 カスタム・テクノロジー株式会社
(22) 出願日	平成10年8月3日(1998.8.3)		神奈川県横浜市港北区新横浜 3-18-1
(65) 公開番号	特開2000-50281(P2000-50281A)		4
(43) 公開日	平成12年2月18日(2000.2.18)	(74) 代理人	100099324 弁理士 鈴木 正剛
審査請求日	平成17年7月20日(2005.7.20)	(72) 発明者	饗農 孝彦 東京都目黒区東山2丁目2番6号 カスタム・テクノロジー株式会社内
		(72) 発明者	竹内 俊一 東京都目黒区東山2丁目2番6号 カスタム・テクノロジー株式会社内
		(72) 発明者	指田 信行 東京都目黒区東山2丁目2番6号 カスタム・テクノロジー株式会社内
			最終頁に続く

(54) 【発明の名称】 動きベクトル検出方法および装置、記録媒体

(57) 【特許請求の範囲】

【請求項1】

コンピュータプログラムにより動作する装置が実行する動きベクトル検出方法であって

連続する第1画像フレームと第2画像フレームとの間の対応画素を指し示す動きベクトルを検出する際に、前記第2画像フレームを、複数のブロックに分割するとともに各ブロックを互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する工程と、

第1グループに属するすべてのブロックについて動きベクトル候補を画像フレーム間の平均誤差評価関数値に基づいて特定する工程と、

第2グループに属するすべてのブロックについて個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する工程と、

第1グループに属するすべてのブロックについて個々のブロックに隣接する他ブロックの動きベクトルの平均ベクトル成分を導出し、この平均ベクトル成分を当該ブロックにおける新たな動きベクトル候補として保持する工程とを含み、

保持されている各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする動きベクトル検出方法。

【請求項2】

コンピュータプログラムにより動作する装置が実行する動きベクトル検出方法であって

10

20

連続する第1画像フレームと第2画像フレームとを取得し、前記第1画像フレームを保持するとともに、前記第2画像フレームを複数のブロックに分割し、これにより得られた各ブロックを、互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する前処理工程と、

前記第1グループに属する個々のブロックの画素と前記第1画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持するとともに、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する初期探索工程と、

10

前記第1グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件を満たす場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第1詳細探索工程と、

この第1詳細探索工程後、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件を満たす場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第2詳細探索工程とを含み、

更新された各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする動きベクトル検出方法。

20

【請求項3】

前記初期探索工程は、前記第1グループの個々のブロックと前記第1画像フレームとの間で動きの無い位置を起点として段階的に探索位置を拡げて前記平均誤差評価関数値を演算し、当該平均誤差評価関数値が収束した位置を指し示すベクトル成分を当該ブロックの動きベクトル候補として保持することを特徴とする、

請求項2記載の動きベクトル検出方法。

【請求項4】

前記第1詳細探索工程は、前記初期探索工程により保持されている前記平均ベクトル成分が指し示す第1画像フレーム上の探索位置を所定方向に変えながら画像フレーム間の画素の平均誤差評価関数値を演算し、この平均誤差評価関数値が相対的に小さくなる位置を指し示すベクトル成分を当該ブロックの新たな動きベクトル候補として更新する、

30

請求項3記載の動きベクトル検出方法。

【請求項5】

前記第1詳細探索工程は、前記探索位置を変える過程で、前記平均ベクトル成分が、同一ブロックについて保持されている動きベクトル候補と近似しているかどうかを所定の閾値との比較により判定し、近似している場合は、さらに、その探索位置での平均誤差評価関数値を求めて、その値が当該時点で正常な動きベクトル候補となり得るものが見つかったかどうかを判定するための収束基準値を越えているかどうかを判定し、越えない場合に、当該探索位置を指し示すベクトル成分を前記新たな動きベクトル候補とすることを特徴とする、

40

請求項4記載の動きベクトル検出方法。

【請求項6】

前記第1詳細探索工程は、前記平均ベクトル成分が同一ブロックについて保持されている動きベクトル候補と近似していない場合は、当該平均ベクトル成分が指し示す位置を、前記探索位置の新たな起点とすることを特徴とする、

請求項5記載の動きベクトル検出方法。

【請求項7】

連続する第1画像フレームおよび第2画像フレームを取得する画像フレーム取得手段と

50

前記第 2 画像フレームを複数のブロックに分割するとともに、分割された各ブロックを、互いに他グループのものが隣接する第 1 グループおよび第 2 グループのブロック群に分類する前処理手段と、

前記第 1 グループに属する個々のブロックの画素と前記第 1 画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持するとともに、前記第 2 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する初期探索手段と、

前記第 1 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第 1 詳細探索手段と、

前記第 2 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第 2 詳細探索手段とを有し、

前記第 1 詳細探索手段および前記第 2 詳細探索手段で更新された各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする、

動きベクトル検出装置。

【請求項 8】

前記第 1 詳細探索手段および前記第 2 詳細探索手段の動作タイミング並びに前記所定の条件を変更可能に設定する制御手段をさらに有することを特徴とする、

請求項 7 記載の動きベクトル検出装置。

【請求項 9】

動画像データを画像フレーム単位に分解する画像データ処理装置と、この画像データ処理装置から時系列に出力される第 1 画像フレームと第 2 画像フレームとから動きベクトルを検出する動きベクトル検出装置とを有し、この動きベクトル検出装置で検出された動きベクトルに基づいて前記第 1 画像フレームの動き検出を行って予測符号化を行う装置であって、

前記動きベクトル検出装置が請求項 7 又は 8 記載の動きベクトル検出装置であることを特徴とする予測符号化装置。

【請求項 10】

連続する第 1 画像フレームと第 2 画像フレームとを取得し、前記第 1 画像フレームを保持するとともに、前記第 2 画像フレームを複数のブロックに分割し、これにより得られた各ブロックを、互いに他グループのものが隣接する第 1 グループおよび第 2 グループのブロック群に分類する処理、

前記第 1 グループに属する個々のブロックの画素と前記第 1 画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持する処理

、前記第 2 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する処理、

前記第 1 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する処理、及び、

第 2 グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する処理、をコンピュータ装置に実行させ、更新された各ブロックの動きベクトル候補を前記動きベクトルとし

10

20

30

40

50

て扱うためのプログラムコードが記録されたコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、動画像データの符号化技術に係り、特に、動画像データに含まれる動きベクトルを効率的に検出する方法およびその関連技術に関する。

【0002】

【発明の背景】

例えばテレビ会議システムのように、動画像データを通信媒体を通じて相手側端末に伝送したり、動画像データをDVD (Digital Video Disk) 等に記録する際に、符号化を施し、原データサイズ以下に圧縮することが一般的になっている。

10

【0003】

通常、動画像データは多数の画像フレームから構成されるが、時間的な相関を利用すれば、これを効率的に符号化することが可能であることは、良く知られている。例えば、時刻 t_1 と時刻 t_2 とで撮像して得た画像フレーム # 1 と画像フレーム # 2 とが存在する場合に、その差分データである画像フレーム # 1 2 を計算によって求め、この画像フレーム # 1 2 を画像フレーム # 2 の代わりに符号化することで、効率良く動画像データを圧縮することが可能となる。これは、画像フレーム # 1 2 が画像フレーム # 1 と画像フレーム # 2 との時間的な相関を利用したデータであり、通常の画像の動き程度では小さな値をもつことが多いことによる。

20

【0004】

最近は、効率的な符号化技術を用いて圧縮率をより向上させるための手法も種々提案されている。

例えば、画像フレーム間の動きベクトルを検出し、この動きベクトルを符号化することで圧縮率を高める技術が知られている。この技術は、例えば画像フレーム # 2 を、微小な領域、例えば 16×16 画素の矩形ブロックに分割し、各矩形ブロックを画像フレーム # 2 に透間なく並べる。そして、それぞれの矩形ブロックに似た小領域を画像フレーム # 1 から探し出し、画像フレーム # 1 からの、その小領域における差分データを計算する。

【0005】

上記の差分データとその絶対値の平均値とを演算する。この絶対値の平均値は、画像フレーム # 1 2 のそれよりも更に小さくなって、符号化効率を向上させることが期待できる。この場合、探し出した小領域の位置からのオフセット値は、「動きベクトル」として、差分データと共に符号化する必要があるが、それでも符号化後のデータサイズは、動きベクトルを使用しない場合に比べて小さくなるのが殆どである。

30

【0006】

このように、画像フレーム # 1 から画像フレーム # 2 の矩形ブロックに似た画素特徴をもつ小領域を探し出し、両者のオフセットを検出することを、「動きベクトル検出」と呼ぶ。具体的には、画像フレーム # 2 上の矩形ブロック G_2 に似た画素特徴をもつ画像フレーム # 1 上の位置 G_1 を探索し、その位置 G_1 の画素値と矩形ブロック G_2 に属するすべての画素値との評価関数値の合計を求め、それが最小となる位置を指し示すベクトル成分を動きベクトル m_b とする。通常、評価関数値には、絶対値や差分の 2 乗値が用いられる。このように動きベクトルを使用して動画像データを圧縮する手法として、CCITT 勧告による H. 261 方式や、ISO による MPEG (Moving Picture Experts Group) 方式等が知られている。

40

【0007】

ところで、動きベクトル検出に際しては、ある画像フレームの矩形ブロックに対応する他の画像フレーム上の位置をどのようにして特定するか、つまり探索手法が問題となる。これについては、良く知られた手法として、「全探索法」と呼ばれる探索手法がある。

【0008】

全探索法では、2つの画素がどの程度似ているかを誤差評価関数で表し、その合計値が最

50

も小さい画素を含む矩形ブロックを似ている矩形ブロックとする。誤差評価関数は、 $ferror(pixel1, pixel2)$ 、例えば画素値 $pixel1$ と画素値 $pixel2$ の差分の絶対値 $|pixel1-pixel2|$ 、または、差分の2乗 $((pixel1-pixel2) * (pixel1-pixel2))$ で表される。いま、符号化しようとする画像フレームの、ある位置 (x, y) における矩形ブロックをAとして、この矩形ブロックAのそれぞれの画素をA (bx, by) とする。一方、動きベクトル検出の際の探索範囲となる画像フレームの、ある位置 $(x+dx, x+dy)$ における矩形ブロックAと同じ大きさの画素集合をBとして、それぞれの画素をB (bx, by) とする。矩形ブロックAの画素集合Bに対するそれぞれの画素の誤差評価関数の合計値をE (dx, dy) とすると、この合計値E (bx, by) は、下式で表される。

$$E(dx, dy) = \sum_{bx, by} (ferror(A(bx, by), B(bx, by)))$$

10

dx の値が $dx1..dx2$ 、 dy の値が $dy1..dy2$ をとり得るとして、この $(dx1, dy1)..(dx2, dy2)$ の矩形を探索範囲とすると、全探索法は、この探索範囲内で合計値E (dx, dy) が最小になる (dx, dy) を求めることで、動きベクトルを検出する。

【0009】

全探索法は、誤差評価関数を探索範囲内ですべての位置において演算することから、動きベクトル検出が正確であると推測されるが、以下のような問題もある。

【0010】

まず、演算量が膨大になるという問題がある。

より激しい動きに対応するためには、探索範囲を拡げる必要がある。つまり、画像フレーム間の動きベクトルの長さが大きいときに探索範囲が狭いと、真の動きベクトルの方向を正しく検出できないので、これを正しく検出するためには、探索範囲を拡げる必要がある。しかし、そうすると、演算量が膨大となってしまう。

20

【0011】

また、探索結果が探索順序に依存するという問題もある。

つまり、誤差評価関数値の合計E (dx, dy) は、同一探索範囲内の異なる場所で同じ値をとり得るため、探索範囲内での探索順序に結果が依存する。その結果、真の動きを反映しない動きベクトルを検出することがある。特に、画像テクスチャが平坦なものであったり、繰り返しパターンであった場合は、ブロックの誤差評価関数値の合計は、異なる場所で似たような値をとることが多い。この場合は、本来の動きとは、かけ離れた位置に、たまたま参照元の矩形ブロックと似ている画素集合を発見する場合がある。この場合は、検出された動きベクトルが真の動きを反映しない。

30

【0012】

さらに、全探索法では、画像の単純な平行移動を仮定し、また、ノイズも無いとして探索するが、現実の動画は絶えず変形を伴っており、また、画像フレームにノイズが混入する場合がある。この場合も、最小の平均誤差評価関数を与える動きベクトルが、必ずしも真の動きを反映しない。動きの激しい画像の動きを捉えるために探索範囲を拡げると、このような誤検出が発生する機会も増加する。この場合、著しく長い誤った動きベクトルとなってしまうことがある。動画データの符号化の際には、動きベクトルも符号化の対象になるので、このような場合には、誤差は小さいけれども符号化ビット量が多い動きベクトルとなってしまう、結果的に符号化効率の低下をもたらす。

40

【0013】

演算量が多いこの全探索法では、探索範囲を拡げて動きベクトルを正しく探索することが困難となるため、演算量を削減できる手法、例えば、「画素間引き法」、「探索位置間引き法」、「縮小画像を用いた階層探索法」も実用化のための検討がなされている。以下、これらの手法を簡単に説明する。

【0014】

(1) 画素間引き法

画素間引き法は、矩形ブロックにおける平均評価関数を算出する過程において、矩形ブロック内のすべての画素を用いずに、代表点だけを用いる手法である。例えば、矩形ブロックの大きさを 16×16 画素とした場合、矩形ブロック内の全画素(256画素)を用い

50

ないで、その半分の128画素を対象にしたり、全画素の1/4である64画素を対象とする。この手法では、対象となる画素が間引かれるため、全探索法に比べれば動きベクトルの検出精度は劣る。特に、間引いた画素における誤差評価関数の値がその矩形ブロックの代表画素の平均に比較して著しく異なる場合には、誤検出が起こる。

【0015】

(2) 探索位置間引き法

探索位置間引き法は、探索範囲から風潰しに平均誤差評価関数を求めずに、まず、粗い探索範囲で動きベクトル候補の検出を行い、その後、順次精度を上げていく手法である。例えば、最初に4画素おきに探索して動きベクトル候補Aを求め、次に、動きベクトル候補Aを中心にして、2画素おきに探索して動きベクトル候補Bを求める。最後に動きベクトル候補Bの周囲の画素を探索して最終的な動きベクトルCを得る。

10

【0016】

この手法は、エッジ部分の画素の空間における変化を見逃してしまう可能性があるため、全探索法に比べれば動きベクトルの検出精度は劣る。また、粗い探索の時点で動きベクトル候補を限定してしまうため、精度を上げていく過程において、一旦誤った動きベクトル候補を決めてしまうと、それ以降は、それを基準にして探索を行うため、正しい動きベクトルを検出することができない。

探索位置間引き法と画素間引き法の併用も可能である。この場合、演算量は更に削減されるが、誤検出が生ずる可能性も高くなる。

【0017】

(3) 縮小画像を用いた階層探索法

縮小画像を用いた階層探索法は、縮小画像を用いて動きベクトル候補の検出を行い、その後、画像の縮小率を低くしていった順次精度を上げていく手法である。例えば、まず、原画像を1/4に縮小した縮小画像で動きベクトルAを求める。次いで、原画像を1/2に縮小した画像において、動きベクトルAと相対的に同じ位置を中心にして他の動きベクトルBを求める。最後に、原画像において動きベクトルAと相対的に同じ位置を中心にして最終的な動きベクトルCを得る。この手法は、演算量の観点からは、探索位置間引き法と画素間引き法を併用したものと同等である。縮小画像においては、探索位置と画素のそれぞれが間引きされるからである。特に、縮小画像生成に単なる間引きを適用した場合、行われる演算量は、探索位置間引き法と画素間引き法を併用した場合と全く等価になる。

20

30

【0018】

この手法では、原画像を縮小する際に画像の細かいテクスチャが失われるので、細かい絵柄を含む画像に対しては全探索法に比べて動きベクトルの検出精度は劣る。また、動きベクトルの検出精度は縮小画像の生成方法に依存する。単純な間引きではなく、例えば、1/2縮小画像を、これに対応する原画像の2x2画素、もしくはそれ以上の画素を参照して重み付けを行い、その平均をとることによって作成した場合は、探索位置間引き法と画素間引き法を併用したものよりは、動きベクトルの検出精度は高まる。これは、画素間引き法が単純に間引いた画素を捨て去るのと比べて、縮小画像生成時に全ての画素を参照することによる。また、探索点間引き法に対しても同様のことがいえ、探索点間引き法が見逃してしまう可能性のあるエッジ等の情報も捉らえる確率は高くなる。これは、探索点間引き法が単純にその探索位置を無視したのに比べて、縮小画像を用いた階層探索法は、縮小画像を生成する過程において全ての探索位置を隙間なく調べるからである。

40

【0019】

また、探索位置間引き法と同様、粗い探索点でまず動きベクトル候補を限定するため、必ずしも最小の平均誤差関数値を与える動きベクトルが求まるとは限らない。また、精度を上げていく過程において、一旦誤った動きベクトルを求めてしまうと、それ以降は、正しい動きベクトルを検出できなくなる。

なお、縮小画像における階層探索法の場合も、探索位置間引き法または画素間引き法、あるいは、その両方を併用することも可能であり、その場合、演算量は更に削減されるが、動きベクトル検出精度における個々の利点は失われ、誤った動きベクトルを検出する可能

50

性が高くなる。つまり、演算量と検出精度とは、トレード・オフの関係にある。

【0020】

いずれにしても、動きベクトルの検出に際して、一定の検出精度を確保しようとするれば、かなりの演算量が必要となる。そのため、これまでは、演算処理を並列的に実行できるようにした専用LSIが用いられることが多く、CPUやMPUに読み込まれて実行されるソフトウェアで動きベクトルを検出することは殆ど行われていない。しかしながら、CPU等の処理速度は年々高速化しているため、今後は、柔軟な処理形態が可能となるソフトウェアによる実装も考えられる。動きベクトル検出のすべてをソフトウェア化しないまでも、階層的に動きベクトルを検出する際に、一部の層、例えば、最終的に原画像データで1画素おきに探索する部分だけをハードウェアによる実装として、残りをソフトウェア化

10

【0021】

現在、ソフトウェアによって動きベクトル検出を行う場合の障害は、画像フレーム間の画素集合の対応位置を見つけるための演算量が膨大なために処理の実時間性が保証できず、逆に実時間性を重視した場合には検出精度が著しく低下することである。特に全探索法を用いて実時間性を保証しようとする、非常に狭い探索範囲しか実現できない。

【0022】

そこで、本発明の課題は、上述の各探索法の利点を活かして、広い探索範囲を確保しつつ誤った動きベクトルの検出を抑制することができる、改良された動きベクトル検索技術を提供することにある。また、ソフトウェアによる実装をも容易にする改良された手法を提供することにある。

20

【0023】

【課題を解決するための手段】

本発明は、上記課題を解決する、改良された動きベクトル方法を提供する。

この動きベクトル検出方法は、コンピュータプログラムにより動作する装置が実行する動きベクトル検出方法であって、連続する第1画像フレームと第2画像フレームとの間の対応画素を指し示す動きベクトルを検出する際に、前記第2画像フレームを、複数のブロックに分割するとともに各ブロックを互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する工程と、第1グループに属するすべてのブロックについて動きベクトル候補を画像フレーム間の平均誤差評価関数値に基づいて特定する工程と、第2グループに属するすべてのブロックについて個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する工程と、第1グループに属するすべてのブロックについて個々のブロックに隣接する他ブロックの動きベクトルの平均ベクトル成分を導出し、この平均ベクトル成分を当該ブロックにおける新たな動きベクトル候補として保持する工程とを含み、保持されている各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする。

30

これにより、第2画像フレームのすべてのブロックについて平均誤差評価関数値を演算する場合に比べて、演算量を少なくすることができる。

【0024】

本発明が提供する他の動きベクトル検出方法は、コンピュータプログラムにより動作する装置が実行する動きベクトル検出方法であって、連続する第1画像フレームと第2画像フレームとを取得し、前記第1画像フレームを保持するとともに、前記第2画像フレームを複数のブロックに分割し、これにより得られた各ブロックを、互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する前処理工程と、前記第1グループに属する個々のブロックの画素と前記第1画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持するとともに、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する初期探索工程と

40

50

、前記第1グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件を満たす場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第1詳細探索工程と、この第1詳細探索工程後、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件を満たす場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第2詳細探索工程とを含み、更新された各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする。

【0025】

上記他の動きベクトル検出方法における、ある実施の態様では、前記初期探索工程は、前記第1グループの個々のブロックと前記第1画像フレームとの間で動きの無い位置を起点として段階的に探索位置を拡げて前記平均誤差評価関数値を演算し、当該平均誤差評価関数値が収束した位置を指し示すベクトル成分を当該ブロックの動きベクトル候補として保持することを特徴とする。

また、ある実施の態様では、前記第1詳細探索工程は、前記初期探索工程により保持されている前記平均ベクトル成分が指し示す第1画像フレーム上の探索位置を所定方向に変えながら画像フレーム間の画素の平均誤差評価関数値を演算し、この平均誤差評価関数値が相対的に小さくなる位置を指し示すベクトル成分を当該ブロックの新たな動きベクトル候補として更新することを特徴とする。

また、ある実施の態様では、前記第1詳細探索工程は、前記探索位置を変える過程で、前記平均ベクトル成分が、同一ブロックについて保持されている動きベクトル候補と近似しているかどうかを所定の閾値との比較により判定し、近似している場合は、さらに、その探索位置での平均誤差評価関数値を求めて、その値が当該時点で正常な動きベクトル候補となり得るものが見つかったかどうかを判定するための収束基準値を越えているかどうかを判定し、越えない場合に、当該探索位置を指し示すベクトル成分を前記新たな動きベクトル候補とすることを特徴とする。この場合において、前記第1詳細探索工程は、前記平均ベクトル成分が同一ブロックについて保持されている動きベクトル候補と近似していない場合は、当該平均ベクトル成分が指し示す位置を、前記探索位置の新たな起点とすることを特徴とする。

【0026】

本発明は、また、動きベクトル検出装置を提供する。

本発明の動きベクトル検出装置は、連続する第1画像フレームおよび第2画像フレームを取得する画像フレーム取得手段と、前記第2画像フレームを複数のブロックに分割するとともに、分割された各ブロックを、互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する前処理手段と、前記第1グループに属する個々のブロックの画素と前記第1画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持するとともに、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する初期探索手段と、前記第1グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第1詳細探索手段と、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する第2詳細探索手段とを有し、前記第1詳細探索手段および前記第2詳細探索手段で更新された各ブロックの動きベクトル候補を前記動きベクトルとすることを特徴とする。

ある実施の態様では、前記第1詳細探索手段および前記第2詳細探索手段の動作タイミング並びに前記所定の条件を変更可能に設定する制御手段をさらに備えた動きベクトル検

10

20

30

40

50

出装置とする。

【 0 0 2 7 】

本発明は、また、上記の動き検出装置を用いた予測符号化装置を提供する。

この予測符号化装置は、動画像データを画像フレーム単位に分解する画像データ処理装置と、この画像データ処理装置から時系列に出力される第1画像フレームと第2画像フレームとから動きベクトルを検出する本発明の動きベクトル検出装置とを有し、この動きベクトル検出装置で検出された動きベクトルに基づいて前記第1画像フレームの動き検出を行って予測符号化を行う装置である。

【 0 0 2 8 】

本発明は、さらに、コンピュータ読取可能な記録媒体を提供する。

この記録媒体は、連続する第1画像フレームと第2画像フレームとを取得し、前記第1画像フレームを保持するとともに、前記第2画像フレームを複数のブロックに分割し、これにより得られた各ブロックを、互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類する処理、前記第1グループに属する個々のブロックの画素と前記第1画像フレームの探索位置の画素との間の平均誤差評価関数値を演算し、この平均誤差評価関数値が最小となる探索位置を指し示すベクトル成分を当該ブロックにおける動きベクトル候補として保持する処理、前記第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を当該ブロックにおける動きベクトル候補として保持する処理、前記第1グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する処理、及び、第2グループに属する個々のブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分を導出し、この平均ベクトル成分が所定の条件に適合する場合に、当該平均ベクトル成分を当該ブロックの新たな動きベクトル候補として更新する処理、をコンピュータ装置に実行させ、更新された各ブロックの動きベクトル候補を前記動きベクトルとして扱うためのプログラムコードが記録されたコンピュータ読取可能な記録媒体である。

【 0 0 2 9 】

【 発明の実施の形態 】

以下、本発明を、MPEG方式のように、他の画像フレームとの相関を利用して動画像データを符号化する予測符号化に応用した場合の実施形態を説明する。

【 0 0 3 0 】

図1は、本実施形態による予測符号化装置の要部構成図である。

この予測符号化装置1では、入力された動画像データを画像処理装置2で画像フレーム単位に分解し、各画像フレームのデータ(画素値)を、本発明の動きベクトル検出装置3に順次送る。動きベクトル検出装置3は、現在の画像フレームのデータ(以下、「現フレームデータ」と称する)と直前の画像フレームのデータ(以下、「参照フレームデータ」と称する)とを比較して動きベクトルを検出し、検出した動きベクトルを動き補償回路4へ送る。また、現フレームデータをそのまま減算器5へ送る。

動き補償回路4はフレームメモリを有しており、このフレームメモリに保持されている参照フレームデータの動き成分を上記動きベクトルで補償するようになっている。なお、ここでいう「補償」とは、符号化のための動き成分の補償をいい、後述する差分データは含まない。動き成分が補償された参照フレームデータは、減算器5と加算器6へ送られる。減算器5は、現フレームデータと動き成分が補償された参照フレームデータとの減算を行い、その差分データをDCT(Discrete Cosine Transform)回路7へ送る。DCT回路7は、この差分データをDCT変換して量子化器8へ送る。量子化器8は、DCT変換された差分データを量子化して可変長符号器9へ送る。これにより符号化データが出力される。

【 0 0 3 1 】

DCT変換された差分データは、量子化器8へ送られるときに、逆量子化回路10と逆D

10

20

30

40

50

CT回路11にも入力されて元の差分データに復元され、加算器6に入力されるようになっている。加算器6は、上記参照フレームデータと復元された差分データとを加算する。これにより、現フレームデータが復元される。この復元された現フレームデータは、次の参照フレームデータとして、動き補償回路4のフレームメモリに保持される。このように、予測符号化装置1では、現フレームデータと、動きベクトルによってその動き成分が補償された参照フレームデータとの差分データが可変長符号化される。

【0032】

次に、本発明の動きベクトル検出装置3の構成例を説明する。この動きベクトル検出装置3は、後述するように、演算量を従来のものより大幅に低減することができるため、専用LSIによって実現しても良いが、本実施形態では、これをソフトウェアによって実現する。

10

【0033】

すなわち、オペレーティングシステム(OS)が搭載されたコンピュータシステムに、所定のプログラムコードを読み込ませて実行させることにより、図2に示すような機能ブロックを当該コンピュータ装置上に形成し、これをもって、動きベクトル検出装置3を実現する。

【0034】

プログラムコードは、通常は記録媒体に記録され、装置構築時にコンピュータシステムの記憶装置に一括で読み込まれるのが通常であるが、運用中に、適宜、記録媒体から上記記憶装置に読み込まれるような形態も可能である。記録媒体としては、半導体メモリ、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、DVD、磁気テープのほか、通信路を介してそのシステムがアクセス可能なプログラムサーバ等も含まれる。

20

【0035】

また、上記プログラムコードを実行することによって上記各機能が実現されるだけでなく、そのプログラムコードの指示に基づいてコンピュータシステム上で稼働しているOSが実際の処理の一部を行い、その処理を通じて上記各機能が実現される場合も、本発明の範囲である。

コンピュータシステムとしては、スタンドアロン型のコンピュータ装置、分散処理型の複数のコンピュータ装置、ネットワーク環境下で通信可能な複数のコンピュータ装置のいずれであっても良い。

30

【0036】

図2を参照すると、本実施形態の動き検出装置3は、上記コンピュータシステムが、画像フレーム取得部31、ブロック化部32、ブロック分類部33、第1初期探索部34、第2初期探索部35、第1詳細探索部36、第2詳細探索部37、学習制御部38の機能ブロックを有するように構成される。第1初期探索部34、第2初期探索部35、第1詳細探索部36、第2詳細探索部37、学習制御部38で、動きベクトル学習手段を形成している。

【0037】

画像フレーム取得部31は、時間的に連続するフレームデータを取得するものであり、ブロック化部32は、現フレームデータを複数のブロックに分割するものである。ブロック分類部33は、分割された各ブロックを、互いに他グループのものが隣接する第1グループおよび第2グループのブロック群に分類するものである。

40

図3は、分類後の現フレームデータの構造を示した図である。図中、「B0」は第1グループに分類されるブロック、「B1」は第2グループに分類されるブロックである。図示のように、本実施形態では、上下左右に他のグループのブロックが出現するように分類している。

【0038】

第1初期探索部34は、第1グループに属する個々のブロックについて、それに対応する参照フレームデータの該当位置を探索し、個々のブロックと探索位置とを結ぶ動きベクトル

50

ル候補を特定するものである。

【0039】

具体的には、図5に示すように、最初のブロックを用意し、そのブロックと参照フレームデータとの間で動きの無い位置を探索し、その位置を起点(0, 0)として平均誤差評価関数値を求める(ステップS101, S102)。このときの参照フレームデータの位置を指し示すベクトル成分を(0, 0)とする(ステップS103)。

予め定めた探索範囲、例えば起点から数画素分拡大した範囲で、予め定めた精度で最初の探索を行い(ステップS104)、その位置で平均誤差評価関数値を求める(ステップS105)。平均誤差評価関数値がこれまでの最小値よりも小さいかどうかを判定し、小さい場合はその探索位置を指し示すベクトル成分を動きベクトル候補として保持しておく(ステップS106: Yes、S107)。探索開始時には保持されている動きベクトル候補がないので、そのときのベクトル成分(0, 0)を保持しておく。

10

【0040】

平均誤差評価関数値がこれまでの最小値よりも大きかった場合、あるいは動きベクトル候補を保持した後は、上記範囲での最後の探索位置になるまで、次の位置にずらして再度ステップS105以降の処理を繰り返す(ステップS106: No、S108: No、S109)。上記範囲での探索が終了した場合は、第1グループの次のブロックについて、ステップS102以降の処理を繰り返す(ステップS110: No、S111)。

【0041】

これにより、第1グループのブロック数と同数の動きベクトル候補を特定することができる。但し、この段階では、おおまかな精度の動きベクトル候補となっており、誤差を多分に含んでいる。

20

【0042】

第2初期探索部35は、第2グループに属する個々のブロックに対応する参照フレームデータの該当位置を探索し、個々のブロックと探索位置とを結ぶ動きベクトル候補を特定するものである。

【0043】

具体的には、図6に示すように、最初のブロックについて、隣接する第1グループのブロックについて既に求められている動きベクトル候補を平均化し、それにより得られたベクトル成分をそのブロックの動きベクトル候補とする(ステップS201, S202)。そして、その動きベクトル候補が指し示す参照フレームデータの位置との間で平均誤差評価関数値を求めておく(ステップS203)。これを第2グループのすべてのブロックについて繰り返す(ステップS204)。これにより、第2グループのブロック数と同数の動きベクトル候補を特定することができる。

30

なお、動きベクトル候補を特定するだけであれば、上記ステップS202の処理は必ずしも必要ではない。

【0044】

第1詳細探索部36は、第1グループに属する個々のブロックに隣接する他ブロック(つまり第2グループのブロック群)の動きベクトル候補の平均ベクトル成分を学習して当該ブロックの新たな動きベクトル候補として更新するものであり、第2詳細探索部37は、第2グループに属する個々のブロックに隣接する他ブロック(つまり第1グループのブロック群)の動きベクトル候補の平均ベクトル成分を学習して当該ブロックの新たな動きベクトル候補として更新するものである。

40

【0045】

第1詳細探索部36における学習処理は、具体的には、図7に示す手順で行われる。前提として、学習のための探索位置の移動ルートは予め定めておく。例えば、探索開始位置を起点にスパイラル状に拡げていくものとする。また、以下の処理は、便宜上、学習処理のための時間制限がない場合の例を示すものである。

【0046】

まず、第1グループに属する最初のブロックについて、上下左右の他ブロックの動きベク

50

トル候補を平均化したベクトル成分を求める（ステップS301, S302）。そして、このベクトル成分が、既に特定されている動きベクトル候補と予め定めた閾値よりも近似しているかどうかを判定する（ステップS303）。これは、ノイズ成分が混入しない場合には、ベクトル成分が以前のものに近似するのに対し、ベクトル成分が混入した場合には、ベクトル成分の方向や長さが以前のものとは大きく異なる点に着目した処理である。例えば、図4上段では、ノイズ成分を表すベクトル成分41が混入している。このベクトル成分41は、他のベクトル成分とは異なった向きと長さを持つのが通常である。そこで、このステップS303で他のベクトル成分との近似性を判定し、近似していない場合は、平均化したベクトル成分を求めることで、このノイズ成分が排除されることになる。図4下段におけるベクトル成分42は、この平均化されたベクトル成分である。

10

【0047】

ステップS303において、ベクトル成分が近似していないと判定した場合は、そのときのベクトル成分が差し示す位置で平均誤差評価関数値を求め、以後の探索において、この位置を探索の起点にする（ステップS303: No, S304）。つまり探索ルートを修正する。

【0048】

ベクトル成分が近似している場合は、その位置での平均誤差評価関数値を求め、それが学習制御部38で定めた「収束基準値」、すなわち、その時点で正常な動きベクトル候補となり得るものが見つかったかどうかを判定するための閾値を越えているかどうかを判定する（ステップS303: Yes, S305）。越えている場合は、正しい動きベクトルが検

20

【0049】

まず、探索位置をそのブロックにおいて、次の位置に設定し、その位置で平均誤差評価関数値を求める（ステップS305: Yes, S306, S307）。その平均誤差評価関数値がこれまでの値の最小値よりも小さければ、その位置を指し示すベクトル成分を、そのブロックの動きベクトル候補として更新する（ステップS308）。

なお、ステップS305において、平均誤差評価関数値が収束基準値を越えていなかった場合は、直ちにステップS309の処理に移行する（ステップS305: No）。これにより、動画像の動きの変化が激しい場合には探索処理のための十分な演算量を確保し、逆に、動きの変化が少ない場合には演算量を削減することが可能となる。

30

【0050】

以上の処理を予め定めた探索範囲において繰り返し（ステップS309: No, S310）、探索範囲のすべての位置を探索した場合は第1グループの次のブロックについて同様の処理を繰り返す（ステップS311: No, S312）。

【0051】

第2グループについても第1グループの場合と同様の処理を行う。つまり、図7において、「第2グループ」とあるのを「第1グループ」と置き換えることで、これを第2グループの処理に読み替えることができる。

【0052】

学習制御部38は、上記第1初期探索部34、第2初期探索部35、第1詳細探索部36、第2詳細探索部37における動作タイミング、動作順序、前述の「閾値」や「収束基準値」の変更等を行うものである。上述の第1詳細探索部36および第2詳細探索部37についての説明は、処理時間に制限が無い場合の例であった。しかし、実際には、演算量と実時間性確保との関係で、処理時間や処理順序には制限を設けている。この場合の各部の動作タイミング等を制御するのが学習制御部38である。

40

【0053】

この学習制御部38による実動作時の制御手順は、例えば図8に従う。すなわち、最初に第1初期探索部34を動作させ、第1グループに属するすべてのブロックについて、概略的ながら動きベクトル候補を特定できるだけの探索範囲で平均誤差評価関数値を演算させる。そして、平均誤差評価関数値が最小となる探索位置を指し示すベク

50

トル成分をそれぞれ当該ブロックにおける動きベクトル候補として特定させる（ステップ S 4 0 1）。

その後、第 2 初期探索部 3 5 を動作させ、第 2 グループに属するすべてのブロックに隣接する他ブロックの動きベクトル候補の平均ベクトル成分をそれぞれ当該ブロックにおける動きベクトル候補として特定させる（ステップ S 4 0 2）。

【 0 0 5 4 】

その後、第 1 詳細探索部 3 6 を、「収束基準値」をやや高めに設定して動きベクトル候補を見つけ易くしておいて一定時間動作させる（ステップ S 4 0 3、S 4 0 4）。これにより、第 1 グループのブロックにおける探索位置が、初期状態から探索ルート上を少し移動する。第 2 詳細探索部 3 7 も、「収束基準値」もやや高めに設定して一定時間動作させる（ステップ S 4 0 5）。これにより、第 2 グループのブロックにおける探索位置が、初期状態から探索ルート上を少し移動する。

10

【 0 0 5 5 】

この時点で、動きベクトル検出の処理を終了させても良いが、時間的に余裕がある場合、つまり処理の許容時間に達していない場合は（ステップ S 4 0 6 : No）、「収束基準値」を前回よりも低くして、つまり、より正確性を増した上で探索ルートを広げさせ、ステップ S 4 0 4 移行の処理を繰り返す（ステップ S 4 0 7）。その後、第 2 詳細探索部 3 7 に対して、第 1 詳細探索部 3 6 と同様の制御を行う（ステップ S 4 0 7）。なお、「収束基準値」の初期値および変更の幅は、対象となる動画データにおける画像の動きの激しさに応じて適宜変えるようにする。

20

【 0 0 5 6 】

これにより、実時間性が重要視される場合と、検出精度が重要視される場合とで、第 1 詳細探索部 3 6 および第 2 詳細探索部 3 7 の動作回数を調整できるようになる。

また、一旦、第 1 初期探索部 3 4 で第 1 グループにおける動きベクトル候補を特定しておけば、第 2 初期探索部 3 5、第 1 詳細探索部 3 6、第 2 詳細探索部 3 7 では、隣接する他ブロックのベクトル成分（動きベクトル候補）の平均ベクトル成分を演算するだけで動きベクトル候補を特定できるので、演算量の大幅な削減が可能になる。さらに、複数の動きベクトル候補の平均をとっているため、突発的な変化成分、例えばノイズ成分の混入に対して頑強であり、精度の良い動きベクトル検出が可能になる。しかも、学習が繰り返されるたびに、これらの突発的な変化成分が吸収されるので、検出精度がより高まる効果がある。

30

【 0 0 5 7 】

また、学習が繰り返されるたびに「収束基準値」が低くなるので、ブロック上の画素特徴と参照フレームデータの位置との一致性がとりやすくなる。

【 0 0 5 8 】

なお、以上の説明は、本発明の動きベクトル検索方法を、全探索法の改良として考えた場合であるが、更に、本発明の動きベクトル検索方式を階層的に用いることで縮小画像を用いた階層探索法にも応用が可能である。この場合、通常の縮小画像を用いた階層探索法と比較して、演算量（演算時間）の低減と、検出精度の向上が期待できるようになる。また、各階層において誤った動きベクトルを訂正する効果が期待できるため、一旦誤りが発生しても、次の階層でこれを補正できる場合がある。

40

【 0 0 5 9 】

また、本発明の動きベクトル検出方法により、予測符号化装置 1 における符号化効率の向上も見込まれる。例えば、縮小画像を用いた階層探索法に適用することで、各階層において空間的に滑らかな動きベクトルにすることができるため、誤った動きベクトルであってもそれが著しく長くはならない。

【 0 0 6 0 】

【 発明の効果 】

以上の説明から明らかなように、本発明によれば、広い探索範囲を確保しつつ誤った動きベクトルの検出を抑制することができるとう、特有の効果がある。

50

また、演算量を低減させることができ、しかも探索処理の実時間性を保証することが可能なので、ソフトウェアによる実装も容易になる。

【図面の簡単な説明】

【図 1】本発明を適用した予測符号化装置の構成図。

【図 2】本実施形態の動き検出装置の機能ブロック構成図。

【図 3】第 1 グループおよび第 2 グループのブロック群に分類した現フレームデータの構造説明図。

【図 4】上段は、ノイズ成分が混入したベクトル成分の説明図、下段は平均化されたベクトル成分の説明図。

【図 5】第 1 初期探索部による処理手順説明図。

10

【図 6】第 2 初期探索部による処理手順説明図。

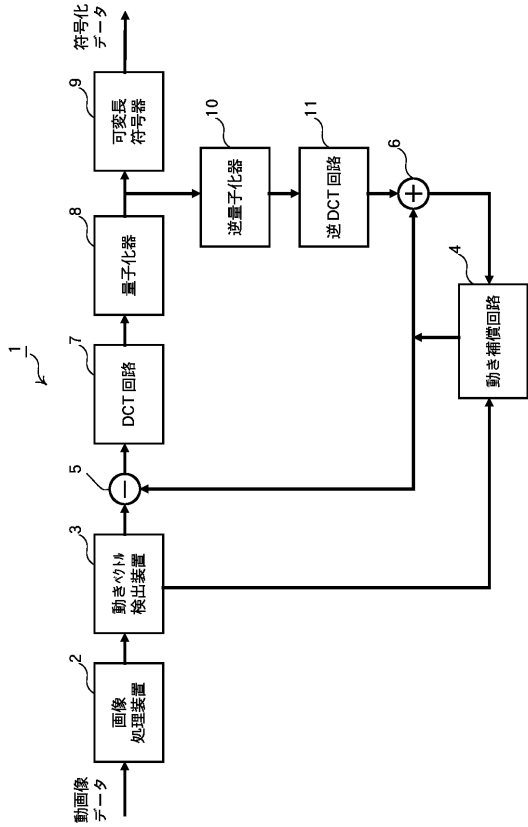
【図 7】第 1 詳細探索部による処理手順説明図。

【図 8】学習制御部における制御手順説明図。

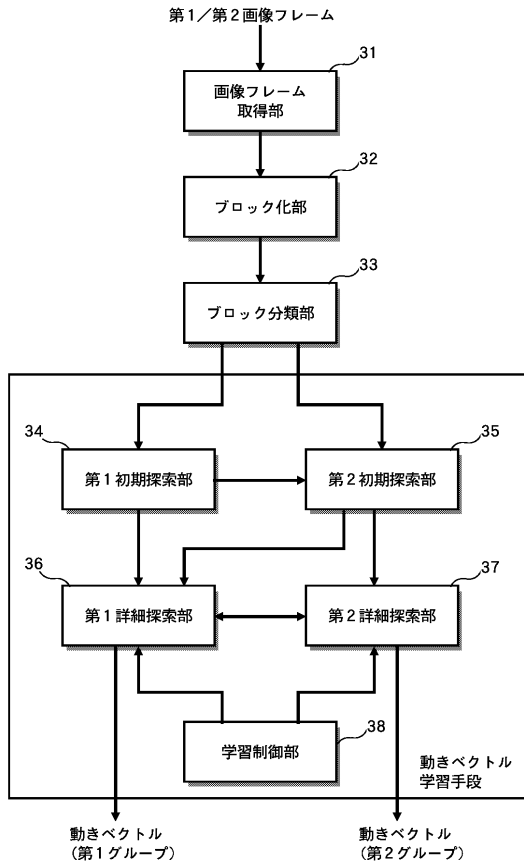
【符号の説明】

- | | | |
|----|------------|----|
| 1 | 予測符号化装置 | |
| 2 | 画像処理装置 | |
| 3 | 動きベクトル検出装置 | |
| 4 | 動き補償回路 | |
| 5 | 減算器 | |
| 6 | 加算器 | 20 |
| 7 | DCT回路 | |
| 8 | 量子化器 | |
| 9 | 可変長符号器 | |
| 10 | 逆量子化回路 | |
| 11 | 逆DCT回路 | |
| 31 | 画像フレーム取得部 | |
| 32 | ブロック化部 | |
| 33 | ブロック分類部 | |
| 34 | 第1初期探索部 | |
| 35 | 第2初期探索部 | 30 |
| 36 | 第1詳細探索部 | |
| 37 | 第2詳細探索部 | |
| 38 | 学習制御部。 | |

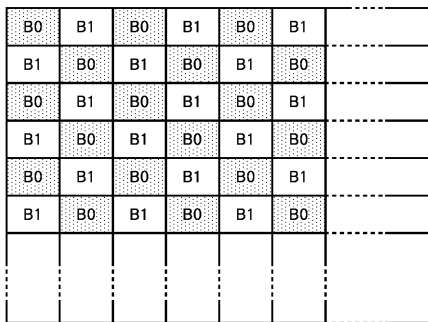
【図1】



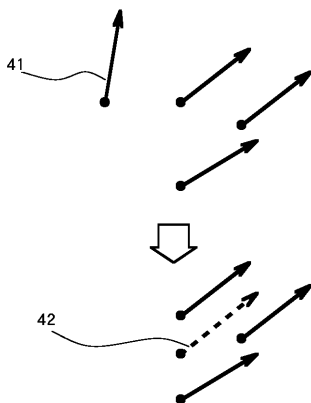
【図2】



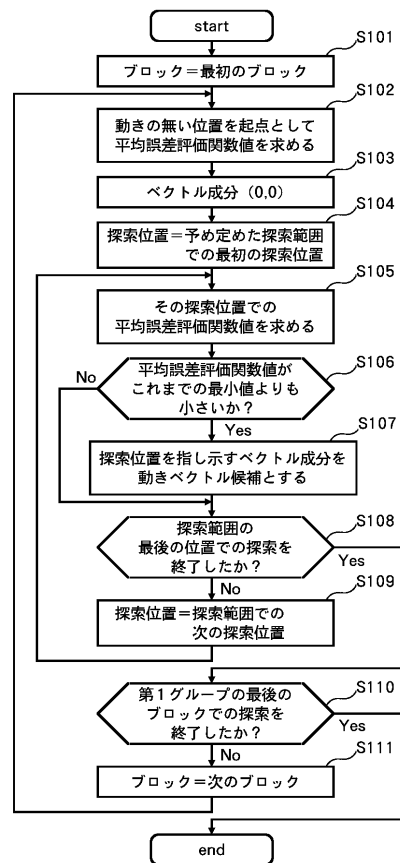
【図3】



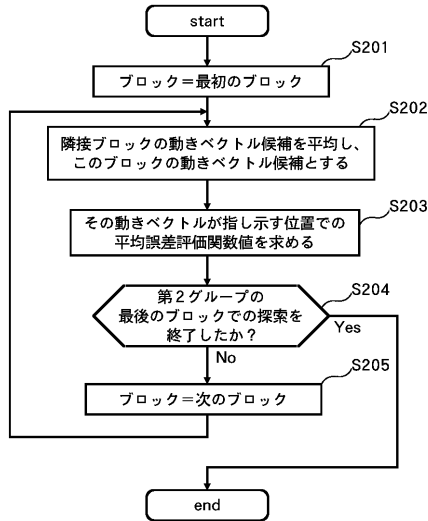
【図4】



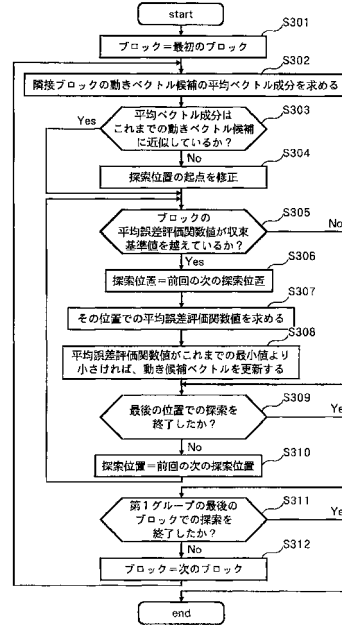
【図5】



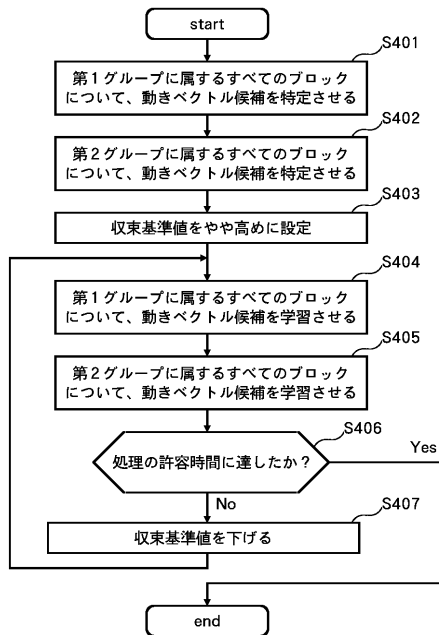
【 図 6 】



【 図 7 】



【 図 8 】



フロントページの続き

審査官 坂東 大五郎

- (56)参考文献 特開平07-177519(JP,A)
特開平05-336510(JP,A)
特開平08-009340(JP,A)
特開平07-162864(JP,A)
特開平09-261646(JP,A)

- (58)調査した分野(Int.Cl., DB名)
H04N 7/24-7/68