

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5163409号
(P5163409)

(45) 発行日 平成25年3月13日 (2013. 3. 13)

(24) 登録日 平成24年12月28日 (2012. 12. 28)

(51) Int. Cl.

F I

H O 4 N 5/232 (2006. 01)

H O 4 N 5/232 Z

H O 4 N 5/225 (2006. 01)

H O 4 N 5/225 A

請求項の数 7 (全 33 頁)

(21) 出願番号 特願2008-258114 (P2008-258114)
 (22) 出願日 平成20年10月3日 (2008. 10. 3)
 (65) 公開番号 特開2010-88085 (P2010-88085A)
 (43) 公開日 平成22年4月15日 (2010. 4. 15)
 審査請求日 平成23年9月22日 (2011. 9. 22)

(73) 特許権者 000002185
 ソニー株式会社
 東京都港区港南1丁目7番1号
 (74) 代理人 100094053
 弁理士 佐藤 隆久
 (72) 発明者 山下 紀之
 東京都港区港南1丁目7番1号 ソニー株
 式会社内
 (72) 発明者 藤田 俊史
 東京都港区港南1丁目7番1号 ソニー株
 式会社内
 審査官 佐藤 直樹

最終頁に続く

(54) 【発明の名称】 撮像装置、撮像方法、およびプログラム

(57) 【特許請求の範囲】

【請求項 1】

合成処理により1枚のパノラマ画像を生成するために、撮像装置を移動させながら光学系を通して複数の被写体像を撮像する撮像素子と、

時間的に隣接する2枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検出する画像信号処理部と、

上記画像信号処理部が、移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す制御部と、

を備える

撮像装置。

【請求項 2】

上記画像信号処理部は、

時間的に隣接する2枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、

上記制御部は、

移動物体と静止被写体の境目では、視差と動被写体を区別することなく警告を発する請求項1記載の撮像装置。

【請求項 3】

上記制御部は、

被写体のどの部分も少なくとも２枚の画像にまたがって写るように重複率を保持して数枚の撮影画像を得るように制御する

請求項１または２記載の撮像装置。

【請求項４】

撮像装置を移動させながら、光学系を通して撮像素子で被写体像を撮像するステップと、

時間的に隣接する２枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検出するステップと、

上記移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促すステップと

を有する撮像方法。

【請求項５】

時間的に隣接する２枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、

移動物体と静止被写体の境目では、視差と移動被写体を区別することなく警告を発する

請求項４記載の撮像方法。

【請求項６】

被写体のどの部分も少なくとも２枚の画像にまたがって写るように重複率を保持して数枚の撮影画像を得るように制御する

請求項４または５記載の撮像方法。

【請求項７】

撮像装置を移動させながら、光学系を通して撮像素子で被写体像を撮像する処理と、

時間的に隣接する２枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検出する処理と、

上記移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す処理と

を有する撮像処理をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、複数の画像を合成する機能を有する撮像装置、撮像方法、およびプログラムに関するものである。

【背景技術】

【０００２】

カメラ一体型ＶＴＲまたはデジタルスチルカメラなどを用いてパノラマ撮影に関する場合、カメラをその都度静止させて撮影するか、もしくは、カメラを動かしながら撮影する場合には、画像のブレをなくすためにカメラを低速で動かす必要がある。

後者の場合、撮影は高速シャッタを用いる必要もある。

【０００３】

これに対して、画像解像度を維持しつつカメラを素早く動かしながら撮影する方法が特許文献１に提案されている。

そこで用いられる技術は、カメラの動作方向と動作角速度を検出するとともに、その動作と逆向きに同角速度で光軸を動作させることにより、画像変化を打ち消し、あたかも１点を注視するかのごとく撮影する技術である。

【０００４】

また、上記の制御方式を実現するために加速度センサや角速度センサを用いる必要がある場合があるが、両センサとその制御用のフィードバック回路を設けなくても光軸を適

10

20

30

40

50

切に制御する方法が特許文献 2 に提案されている。

【 0 0 0 5 】

この場合にはモニタリングシステムとして用いられ、撮影方向の制御に使用されるステップモータのパルス数をカウントし、その係数値に応じて光軸制御を行うものである。

【特許文献 1】特許第 3 9 2 8 2 2 2 号公報

【特許文献 2】特許第 3 9 2 5 2 9 9 号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

また、広画角で高精細の写真を撮るために、カメラを望遠にして一点から方向を変えて複数回撮影する方法においては、通常、三脚に固定したアクティブな雲台で撮影するのが普通であるが、手持ちのカメラでこれを行うことも可能である。

ところが、カメラを手にとって肩を中心にして回して撮影した場合、カメラの視点よりかなり後方を中心として回すことになり、パララックスの影響を強く受ける。

すべてが遠方の景色なら殆ど問題ないが、近距離の被写体が含まれていると隣接画像との位置関係が遠距離の被写体と異なり、正しく合成することができない。

しかし、現行のカメラでは、視差や移動被写体の影響を管理(監視)することができないことから、撮影直後にやり直しが必要か否かの判断を行うことができない。

【 0 0 0 7 】

本発明は、やり直しが必要か否かを撮影直後に判断することが可能で、使い勝手のよい撮像装置、撮像方法、およびプログラムを提供することにある。

【課題を解決するための手段】

【 0 0 0 8 】

本発明の第 1 の観点の撮像装置は、合成処理により 1 枚のパノラマ画像を生成するために、撮像装置を移動させながら光学系を通して複数の被写体像を撮像する撮像素子と、時間的に隣接する 2 枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検出する画像信号処理部と、上記画像信号処理部が、移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す制御部と、を備える。

【 0 0 1 0 】

好適には、上記画像信号処理部は、時間的に隣接する 2 枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、上記制御部は、移動物体と静止被写体の境目では、視差と動被写体を区別することなく警告を発する。

【 0 0 1 1 】

好適には、上記制御部は、被写体のどの部分も少なくとも 2 枚の画像にまたがって写るように重複率を保持して数枚の撮影画像を得るように制御する。

【 0 0 1 2 】

本発明の第 2 の観点の撮像方法は、撮像装置を移動させながら、光学系を通して撮像素子で被写体像を撮像するステップと、時間的に隣接する 2 枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検出するステップと、上記移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促すステップとを有する。

【 0 0 1 3 】

本発明の第 3 の観点は、撮像装置を移動させながら、光学系を通して撮像素子で被写体像を撮像する処理と、時間的に隣接する 2 枚の画像の重複する領域内で複数の動き検出を行って複数の動きベクトルを求め、異なる値のベクトルが存在する場合に視差を検出したものと認識し、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体を検

10

20

30

40

50

出する処理と、上記移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す処理とを有する撮像処理をコンピュータに実行させるためのプログラムである。

【 0 0 1 4 】

本発明によれば、撮像装置を移動させながら撮影した複数枚の撮影画像が画像信号処理部に入力される。

画像信号処理部では、隣接画像間の移動ベクトルの類似性で視差による影響または動被写体が検出され、検出結果が制御部に出力される。

制御部においては、画像信号処理部が、移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す。

10

【発明の効果】

【 0 0 1 5 】

本発明によれば、取り直しが必要か否かを撮影直後に判断することが可能で、使い勝手の向上を図ることができる。

【発明を実施するための最良の形態】

【 0 0 1 6 】

以下、本発明の実施形態を図面に関連付けて説明する。

【 0 0 1 7 】

図 1 は、本発明の実施形態に係る撮像装置としてのカメラ装置の構成例を示すブロック図である。

20

【 0 0 1 8 】

本カメラ装置 1 0 は、たとえば図 2 に示すように、自動あるいは手動により、一点から方向を変えて複数回撮影した多数の画像（図 2 においては、 $16 \times 8 = 128$ 枚）を得ることが可能である。

カメラ装置 1 0 は、これらの多数、たとえば数千枚の画像をしわ寄せなく正確に合成し、いわゆるパノラマ画像を形成することができるよう構成されている。

すなわち、カメラ装置 1 0 は、固定撮像素子、たとえば C M O S イメージセンサ（C I S）を搭載したデジタルカメラを縦または横方向に高速に振り回して撮影した画像を用いてパノラマ画を作成する機能を有している。

【 0 0 1 9 】

30

そして、本実施形態に係るカメラ装置 1 0 は、以下の第 1 ～ 第 5 の特徴的な構成および機能を有する。

【 0 0 2 0 】

第 1 の構成は以下の点である。

カメラ装置 1 0 を移動させながら複数枚の撮像画像を行い、取得した画像を合成することによってパノラマ画像を生成するにあたって、画像を集光するレンズ（シフトレンズ）の光軸をカメラの移動方向と角速度を打ち消すよう制御する。

これにより、カメラが動いていてもあたかも 1 点を注視するがごとく画像を撮影する。

この構成においては、固体撮像素子としては C I S (CMOS Image Sensor) を用い、中心方向の一部のラインに対して上記の制御を行い撮像する。

40

すなわち、一部のラインの露光時間と読み出し時間を加えた期間に上記の光軸制御が行われ、それ以外の時間においては、光軸を中心付近に戻すような制御が行われる。その際のカメラの撮影方向は C I S のラインに対して垂直方向とする。

そして、カメラ装置 1 0 は、C I S の一部を短冊状に切り出し、その部分に対応した光軸制御を行うことにより、カメラが高速に動いても高いフレームレートで解像度を低下させずにパノラマを生成する。

【 0 0 2 1 】

第 2 の構成は以下の点である。

カメラ装置 1 0 は、連続した画像を、画像認識技術で得られたフレームの移動情報と、姿勢センサからの移動情報を用いて空間的に配置する技術を採用している。

50

この画像認識では、情報を得られない部分を姿勢センサ情報で補い、姿勢センサ情報は画像認識の成功確認や失敗時の補助座標として用いる。空間的に配置された画像は1つのパノラマ画像として完成する。

この場合、カメラ装置10は、主として手持ちで略一点から方向を変えながら複数枚撮影するカメラとして構成される。

カメラ装置10は、3軸（または2軸）加速度センサと3軸（または2軸）角速度センサまたはそのいずれかを含む姿勢センサを有する。

カメラ装置10は、撮影された各画像にはどの方向を向いて撮影したかという姿勢の情報を同時に記録し、撮影された複数の画像をその場で1枚に合成する機能を有する。

カメラ装置10は、画像の重複領域を利用してブロックマッチング等の画像認識機能を使って相互の画像の相対位置関係を算出し、各種の姿勢センサのデータによる画像の位置関係を算出する。

10

そして、カメラ装置10は、算出した相対位置関係と画像の位置関係の双方の選択的協調によってより精度の高い画像の相対位置関係を算出する。

その後、カメラ装置10は、各画像の中心が向く方向、パン角（経度）とチルト角（緯度）および、その光軸のまわりに回転するロール角（傾斜）といった各画像の絶対位置関係を特定し、これを初期値として精密自動合成を行う。

【0022】

第3の構成は以下の点である。

カメラ装置10は、連続した画像の記録において、画像認識技術で得られたフレームの移動情報と、姿勢センサからの移動情報を対応する技術を採用している。

20

カメラ装置10は、片方だけでは不明な、画像のピクセル画角、姿勢センサの静止時の値、姿勢センサ値のピクセル画角対応といった情報を算出する。オフセット、ゲイン等のパラメータを有し、これを変更することによって実際の方向とほぼ一致させることができる。

カメラ装置10は、静止時の姿勢のデータを3軸（または2軸）加速度センサが重力の方向となす角としてスタティックに検出し、それを姿勢情報の初期値とする。

カメラ装置10は、カメラの縦方向、横方向の主に回転移動をたとえば3軸角速度センサを時間積分したもので計算し、各画像の撮影された時点の方向データとする。

カメラ装置10は、画像の重複領域を利用してブロックマッチング等の画像認識手段を使って相互の画像の位置関係を算出する。カメラ装置10は、相互の画像の位置関係を演算で求めると同時にその結果が正しいか正しくないかの判定も行う。

30

カメラ装置10は、ここで判定された結果が正しい場合、この情報でパラメータを校正する。

そして、カメラ装置10は、判定された結果が正しくない場合、既に校正されているパラメータによる姿勢センサの値を使って画像を配置していく。

【0023】

第4の構成は以下の点である。

カメラ装置10は、移動体による影響を検出した場合に警告を出して撮影のやり直しを促す機能を有する。

40

カメラ装置10は、移動体の検出に関し、重複率を50%以上として被写体のどの部分も少なくとも2枚の画像にまたがって写るようにし、隣接画像間の移動ベクトルの類似性で視差による影響や動被写体を検出する機能を有する。

すなわち、カメラ装置10は、移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す。

広い範囲の被写体をさっと一振りで複数の短冊状画像を撮影して一枚に合成するカメラ装置10において、近距離の被写体がパララックスの影響をどの程度受けているかを検出し、カメラの視点を中心として撮影し直すことを促す。

【0024】

第5の構成は以下の点である。

50

カメラ装置１０は、スロープ角速度（ユーザがカメラを振り回す速さ）の適正値を知らせる方法と、速すぎる場合に警告を出して撮影のやり直しを促す。

カメラ装置１０は、表示装置１８、たとえばＬＣＤの画面に横軸に時間、縦軸に姿勢センサ（ジャイロセンサ）の出力（スロープ角速度）をグラフにして表示する。水平画角、水平画素数、シャッタ速度が設定されると最大スロープ角速度が決まるので、その６０％～８０％を適正範囲としてグラフに表示する。

【００２５】

以下に、上述した特徴を有するカメラ装置１０のより具体的な構成および機能について説明する。

【００２６】

カメラ装置１０は、光学系１１、撮像素子１２、アナログフロントエンド（ＡＦＥ）回路１３、姿勢センサ１４、ドライバ１５、システムコントローラ１６、メモリ１７、表示装置１８、操作部１９、発音部２０等を含んで構成されている。

【００２７】

光学系１１は、被写体像を撮像素子１２の撮像面に結像させる。

光学系１１は、通常のレンズ１１１、光軸可変素子としてのシフトレンズ１１２、およびメカニカルシャッタ１１３を含んで構成される。

シフトレンズ１１２は、画像を集光する機能に加えて、ドライバ１５の駆動により光軸の方向を変更可能な機能を有している。

【００２８】

撮像素子１２は、ＣＭＯＳ（Ｃｏｍｐｌｙｍｅｎｔａｒｙ Ｍｅｔａｌ Ｏｘｉｄｅ Ｓｅｍｉｃｏｎｄｕｃｔｏｒ）デバイスまたはＣＣＤ（Ｃｈａｒｇｅ Ｃｏｕｐｌｅｄ Ｄｅｖｉｃｅ）で構成される。

本実施形態では、ＣＭＯＳイメージセンサを例として説明する。また、上述した第１の構成では、固体撮像素子としてＣＭＯＳイメージセンサが適用される。

撮像素子１２は、半導体基板上にマトリクス状に配列した光センサにより光学系１１による被写体像を検出して信号電荷を生成し、同信号電荷を垂直信号線や水平信号線を介して読み出して被写体の画像信号を出力する。

撮像素子１２がＣＭＯＳイメージセンサにより形成される場合、電子シャッタとしてグローバルシャッタとローリングシャッタで露光制御が行われる。この露光制御は、システムコントローラ１６により行われる。

【００２９】

ＡＦＥ回路１３は、たとえば撮像素子１２からの画像信号に含まれる固定パターンノイズを除去し、自動利得制御の動作により信号レベルを安定化してシステムコントローラ１６に出力する。

【００３０】

姿勢センサ１４は、カメラ装置１０の姿勢を検出し、検出結果をシステムコントローラ１６に供給する。

姿勢センサ１４は、たとえば３軸加速度センサ１４１と３軸角速度センサ１４２により構成される。

加速度センサ１４１は重力の方向となす角をスタティックに知ることができ、チルト角とロール角を検出することが可能である。しかし、パン角を検出することはできない。

そこで、移動角を知るためには角速度センサ１４２を用いる。これはジャイロセンサとも言われ、回転中の角速度が電圧信号として検出が可能であり、この電圧信号を積分すると角度になる。そして、３軸であることからパン角、チルト角、ロール角が検出できる。

【００３１】

ドライバ１５は、システムコントローラ１６の制御の下、光学系１１のシフトレンズ１１２の光軸を変更する。

【００３２】

システムコントローラ１６は、ＡＦＥ回路１３の出力信号に対して色補正処理、複数画

10

20

30

40

50

像の合成処理、自動露光制御、オートホワイトバランス制御等を行うための回路である。

【0033】

システムコントローラ16は、画像信号処理部161、および制御部としてのマイクロコンピュータ(μ-COM)162を含んで構成される。

【0034】

画像信号処理部161は、一点から方向を変えて複数回撮影した多数の画像をしわ寄せなく正確に合成を行うことができるように構成した精密合成処理部を有している。

そして、図3に示すように、精密合成処理部1611は、第1色補正機能部16111、合成機能部16112、および第2色補正機能部16113を含んで構成されている。

【0035】

画像信号処理部161は、カメラ装置10を移動させながら複数枚の撮像画像を行った場合に、取得した画像を合成することによってパノラマ画像を生成する。

マイクロコンピュータ162は、姿勢センサ14の検出結果に応じて、画像を集光するレンズ(シフトレンズ)の光軸をカメラの移動方向と角速度を打ち消すよう制御する。

マイクロコンピュータ162は、固体撮像素子としてはCMOSイメージセンサを用いた場合に、中一部のラインの露光時間と読み出し時間を加えた期間に上記の光軸制御を行いそれ以外の時間においては、光軸を中心付近に戻すようにドライバ15を制御する。その際のカメラの撮影方向はCMOSイメージセンサのラインに対して垂直方向とする。

そして、マイクロコンピュータ162は、CMOSイメージセンサの一部を短冊状に切り出し、その部分に対応した光軸制御を行うことにより、カメラが高速に動いても高いフレームレートで解像度を低下させずにパノラマを生成するように制御する。

【0036】

マイクロコンピュータ162は、角速度センサ142の検出信号を積分してカメラ装置10の回転角度を算出し、算出された回転角度に応じて、シフトレンズ112の光軸の変化量を制御する。

あるいは、画像信号処理部161は、撮像された隣り合う画像の動き成分を検出し、マイクロコンピュータ162はこの検出された動き成分に応じて光軸の変化量を制御することも可能である。

あるいは、マイクロコンピュータ162は、算出された回転角度および動き成分を用いて光軸の変化量を制御することも可能である。

【0037】

マイクロコンピュータ162は、撮影された各画像にはどの方向を向いて撮影したかという姿勢の情報をメモリ17に記録する。

画像信号処理部161およびマイクロコンピュータ162は、画像の重複領域を利用してブロックマッチング等の画像認識機能を使って相互の画像の相対位置関係を算出し、各種の姿勢センサのデータによる画像の位置関係を算出する。

マイクロコンピュータ162は、算出した相対位置関係と画像の位置関係の双方の選択的協調によってより精度の高い画像の相対位置関係を算出する。

その後、マイクロコンピュータ162は、各画像の中心が向く方向、パン角(経度)とチルト角(緯度)および、その光軸のまわりに回転するロール角(傾斜)といった各画像の絶対位置関係を特定する。

画像信号処理部161は、これを初期値として精密自動合成を行う。

【0038】

マイクロコンピュータ162は、片方だけでは不明な、画像のピクセル画角、姿勢センサの静止時の値、姿勢センサ値のピクセル画角対応といった情報を算出する。マイクロコンピュータ162は、オフセット、ゲイン等のパラメータを有し、これを変更することによって実際の方向とほぼ一致させることができる。

マイクロコンピュータ162は、静止時の姿勢のデータを3軸(または2軸)加速度センサが重力の方向となす角としてスタティックに検出し、それを姿勢情報の初期値とする。

マイクロコンピュータ１６２は、カメラの縦方向、横方向の主に回転移動をたとえば３軸角速度センサ１４２を時間積分したもので計算し、各画像の撮影された時点の方向データとする。

マイクロコンピュータ１６２は、画像の重複領域を利用してブロックマッチング等の画像認識機能を使って相互の画像の位置関係を算出するが、相互の画像の位置関係を演算で求めると同時にその結果が正しいか正しくないかの判定も行う。

マイクロコンピュータ１６２は、ここで判定された結果が正しい場合、この情報でパラメータを校正する。

そして、マイクロコンピュータ１６２は、判定された結果が正しくない場合、既に校正されているパラメータによる姿勢センサの値を使って画像を配置していく。

10

【００３９】

マイクロコンピュータ１６２は、移動体による影響を検出した場合に、表示装置１８あるいは発音部２０により表示あるいは警告音、または両者により警告を出して撮影のやり直しを促す。

マイクロコンピュータ１６２は、移動体の検出に関し、重複率を５０％以上として被写体のどの部分も少なくとも２枚の画像にまたがって写るようにし、隣接画像間の移動ベクトルの類似性で視差による影響や動被写体を検出する。

すなわち、マイクロコンピュータ１６２は、移動体または視差による影響を検出した場合に警告を出して撮影のやり直しを促す。

マイクロコンピュータ１６２は、近距離の被写体がパララックスの影響をどの程度受けているかを検出し、カメラの視点を中心として撮影し直すことを促す。

20

【００４０】

マイクロコンピュータ１６２は、スweep角速度（ユーザがカメラを振り回す速さ）の適正値を知らせ、速すぎる場合に表示装置１８あるいは発音部２０により表示あるいは警告音、または両者により警告を出して撮影のやり直しを促す。

マイクロコンピュータ１６２は、表示装置１８、たとえばＬＣＤの画面に横軸に時間、縦軸に姿勢センサ（ジャイロセンサ）の出力（スweep角速度）をグラフにして表示する。水平画角、水平画素数、シャッタ速度が設定されると最大スweep角速度が決まるので、その６０％～８０％を適正範囲ＲＮＧとして、図４に示すようにグラフに表示する。

操作手順の概略は以下の通りである。

30

[1]操作部１９のスタートボタンを押してカメラを回し、その後スタートボタンを離す。

[2]スタートボタンが押されている間のスweep角速度が、図４に示すように、表示装置１８の画面に表示される。

[3]適正範囲ＲＮＧより遅い場合には警告はないが、一瞬でも範囲を越えて速い場合には警告音が出る。

【００４１】

以下に、上述した第１～第５の構成について具体的に説明する。

なお、第１～第５の構成における主たる制御はシステムコントローラ１６により行われる。

【００４２】

40

[第１の構成]

第１の構成では、固体撮像素子としてＣＭＯＳイメージセンサを適用することからフレーム／フィールドと言った考え方は存在せず、全てのラインを順次読み出すプログレッシブ方式となる。

【００４３】

図５（Ａ）および（Ｂ）は、本実施形態の第１の構成における撮影形態を説明するための図である。

【００４４】

カメラ装置１０の動かし方としては、基本的に図５（Ａ）に示すような鉛直方向、もしくは図５（Ｂ）に示すような水平方向に回転動作することを前提としている。つまり、Ｃ

50

ＭＯＳイメージセンサの読み出しラインに対してカメラを垂直に動作させる。

また、図５の濃い短冊部３０で示されるように、本実施形態では、マイクロコンピュータ１６２は、ＣＭＯＳイメージセンサの撮像範囲のうち中心部から切り出される短冊に対して光軸制御を行っている。

【００４５】

このように短冊撮影を行うことにより以下の利点が得られる。

- (a)パララックスの影響は短冊の幅が狭いほど有利になる。
- (b)ＣＭＯＳイメージセンサの非同時読み出しの影響も短冊の幅が狭いほど有利になる。
- (c)周辺減光の影響にも有利になる。
- (d)レンズ歪の影響にも有利になる。

10

【００４６】

マイクロコンピュータ１６２は、たとえば姿勢センサ１４の検出結果に応じて、画像を集光するレンズ（シフトレンズ）の光軸をカメラの移動方向と角速度を打ち消すよう制御する。

マイクロコンピュータ１６２は、固体撮像素子としてはＣＭＯＳイメージセンサを用いた場合に、中一部のラインの露光時間と読み出し時間を加えた期間に上記の光軸制御を行いそれ以外の時間においては、光軸を中心付近に戻すようにドライバ１５を制御する。

【００４７】

すなわち、図５で示すような短冊部３０が露光されている間、光軸制御が行われる必要がある。

20

【００４８】

図６は、ＣＭＯＳイメージセンサの露光時間と蓄積電荷の読み出し時間、そして光軸制御時間の関係を示す図である。

ＣＭＯＳイメージセンサの各ラインにおいて露光に引き続き電荷の読み出しが行われるが、あるラインについて読み出しが終了した後、次のラインの露光と電荷の読み出しが行われる。この動作を繰り返して短冊部の全ての電荷処理が行われる間、光軸制御が行われる。

たとえば、シャッタ速度が $1/1000$ 秒（つまり露光時間が 1msec ）、短冊幅が 200 ラインの場合、図６の図の読み出し時間は 1.56msec 、光軸制御時間は 2.56msec となる。ちなみに、撮影のフレームレートが 60fps （ 1 画像あたり約 16.66msec ）において、対応する特許文献１の図３における数値を当てはめてみると、 S_{on} は 2.56msec であり、 $S_{off} = 16.66 - 2.56 = 14.1(\text{msec})$ となる。

30

また、光軸制御される限界角度については、特許文献１では $\pm 1.2(\text{deg})$ であるが、本実施形態の場合、たとえば、 $\pm 0.5^\circ$ の範囲で可変であるところ、 0 度～ 0.3 度までの間を使用する。これは可変範囲の最大値の約 60% である。

【００４９】

このように撮影して得られた短冊画像を用いて、図３の精密合成処理部１６１１で画像合成しパノラマ画を生成する。以下、この精密合成処理部１６１１の画像の合成処理について説明する。

【００５０】

40

このように、本実施形態に係るシステムコントローラ１６は、一点から方向を変えて複数回撮影した画像を、色ムラを補正して１枚に精密合成する機能（たとえばソフトウェア）を有している。

以下、本実施形態の精密合成の特徴な機能部分を具体的に説明する。

【００５１】

第１色補正機能部１６１１１は、レンズ歪補正係数などのパラメータを抽出するときは、ひとつの境界に少なくとも３個のブロックマッチング（ＢＭ）を行い、最低４枚の境界について合成を行い、これが正確になるようにレンズ歪補正係数を決めていく。

換言すれば、第１色補正機能部１６１１１は、原画像からレンズ歪み補正係数などのパラメータを抽出する。

50

そして、第1色補正機能部16111は、すべての部分画像に一樣に周辺減光補正、コントラスト強調、彩度強調、ガンマ補正を行う。

【0052】

合成機能部16112は、第1色補正機能部16111でレンズ歪補正係数などのパラメータが決まり、周辺減光補正、コントラスト強調、彩度強調、ガンマ補正を行った後、全ての境界に少なくとも1個（たとえば3個）のBM（ブロックマッチング）を行う。

合成機能部16112は、すべての境界について同時にBM結果の評価を行い、すべての境界の誤差が小さくなるように光軸の方向を更新して誤差を小さくしていき複数の画像の合成を精密に行う。

【0053】

10

第2色補正機能部16113は、合成機能部16112で精密合成された複数の画像のうち、隣接する画像間の色の違いを減少させるため、すべての部分画像に対して独立に行う色（ムラ）補正を行う。

また、第2色補正機能部16113は、隣接する画像間の色の不連続を検知限界以下まで減少させるための色補正を行う。

【0054】

ここで、精密合成処理部1611における精密合成処理の原理的な概念を説明する。

【0055】

本実施形態においては、基本的に、フーリエ解析(Fourier analysis)に基づく位相相関技術(phase correlation technique)を採用している。

20

つまり、空間的関数のシフトがスペクトル領域における位相のみ変化するというフーリエシフト定理に基づいた技術を採用している。

つまり、2つの関数 f_1 および f_2 が次の関係を満足しているとする。

【0056】

[数1]

$$f_2(x, y) = f_1(x + x_t, y + y_t)$$

【0057】

また、次のようなスペクトル特性を有する。

【0058】

[数2]

$$F_2(u, v) = F_1(u, v) \exp(-2i(ux_t + vy_t))$$

30

【0059】

この式は、等価的に、相互電力スペクトル(Cross-Power Spectrum : CPS)を用いて次のように書き換えることができる。

【0060】

【数3】

$$\frac{F_1(u, v)F_2^*(u, v)}{\underbrace{F_1(u, v)F_2^*(u, v)}_{\text{Cross-power spectrum}}} = \exp(2\pi i(ux_t + vy_t)),$$

40

【0061】

ここで、 F_2^* は複素関数 F_2 の共役関数である。

実際、画像は、図7に示すように、2つの画像の相互電力スペクトルのようなビット雑音である。

したがって、相互電力スペクトル(CPS)のピークを探り出し、それからトランスレーション（平行移動）パラメータ (x_t, y_t) を引き出すことが望ましい。

50

【 0 0 6 2 】

図 7 は、相互電力スペクトル(C P S)を用いたトランスレーションにおけるステッチング画像を示す図である。

図 7 (A) は 2 つの画像のステッチングの結果を示している。2 次元のトランスレーションは、図 7 (B) に示すように、相互電力スペクトル(C P S)のピーク検出によって得られる。ここで、相互電力スペクトル(C P S)が読み可能であれば画像は完全に整合する。

ノイズの多い画像は、最高のピークを検出することが困難であることから、いくつかのピークから選択するようにしても良い。

【 0 0 6 3 】

次に、B M (ブロックマッチング) を用いてパラメータを抽出する原理について、図 8 ~ 図 1 4 に関連付けて説明する。

ここで、B M は上述した相互電力スペクトル(C P S)のピークを引き出す機能を含む。

【 0 0 6 4 】

まず、図 8 に示すように、条件のよい 4 枚の画像 I M 0 , I M 1 , I M 2 , I M 3 を選択する。

たとえば左下が第 0 の画像 I M 0 、その右が第 1 の画像 I M 1 、左上が第 2 の画像 I M 2 、その右が第 3 の画像 I M 3 とする。これらの画像 I M 0 ~ I M 3 は、隣接する画像間の境界で重なり部分を含むように配置される。

図 8 において、境界部分に配置した矩形で示すものがブロック B L K である。

このような配置条件で、B M (ブロックマッチング) を行う。

そして、上下左右の 4 つの境界 B D R 0 1 , B D R 0 2 , B D R 1 3 , B D R 2 3 から、レンズ歪み、画角、チルト角などの情報を取り出す。

【 0 0 6 5 】

B M (ブロックマッチング) についてさらに説明する。

B M は、たとえば図 9 に示すように、一つの境界で 3 箇所について行う。

レンズ歪みがあると、B M の結果は図 1 0 に示すように弓なりになる。

チルト角が正しくないと、B M の結果は図 1 1 に示すような、図中の左右方向に傾いたような誤差が生じる。

レンズ歪みの中心が縦にずれていると、図 1 2 に示すように、上下の境界で左右の伸縮が起きる。左右の境界で上下の伸縮が起きているときは、レンズ歪みの中心が横にずれているときである。

上に向くにつれて画像が回転する場合、図 1 3 に示すように、上下方向に傾いたような誤差が生じる。メカニカルなチルト軸に対してカメラが正面を向いていないときに、図 1 3 のようになる。

これらの誤差が最小となるように各種パラメータを決める。

これにより、どこの 4 枚をつないでも、誤差を小さくすることができる。

【 0 0 6 6 】

たとえばファースト位相相関マッチング(fast phase correlation matching)を適用して画像における対応する B M を行う。ベクトル変位 ($x_{i j}$, $y_{i j}$) を得、3 個のブロックの変位の振舞いを解析することにより、各パラメータを定量化することができる。

【 0 0 6 7 】

以上の 4 枚の B M (ブロックマッチング) を行った後、図 1 4 (A) , (B) に示すように、多くの枚数に拡張し、すべての境界について同時に B M 結果の評価を行う。すべての境界の誤差が小さくなるように光軸の方向を更新して誤差を小さくしていき複数の画像の合成を精密に行う。

この場合、基準画像を 1 枚決め、それ以外の画像を平行移動して最も誤差が小さくなる位置に収束させる。

【 0 0 6 8 】

精密合成処理は、以下の基本的処理に従って行う。それを箇条書き的に示す。

平行移動で最適位置を求め、移動する。

この場合、ループを回す。

移動すべき量の総和を示すパラメータ f_{xy} を 0.0 にする。

上下左右（縦横）のすべての画像について行う。

基準画像は動かさない。

B Mの結果、隣接画像との位置関係が得られているのでこれに基づいて動かすべき量を計算する。

その方法は、真上と右隣のものを加え、真下と左隣のものを引いて平均を作り、 $f[y][x].x$, $f[y][x].y$ とする。

その 80 %を現在の画像の中心位置に加えて新しい画像の中心位置とする。

f_{xy} に、すべての画像の移動すべき量の絶対値の総和を入れる。

この移動によって上下の位置関係、左右の位置関係がどう改善されるかを計算する。

f_{xy} はこの移動を繰り返すことにより、だんだん小さくなる性質を持っている。

言い換えると、これ以上移動することができない状態に収束する。

f_{xy} が十分小さくなったら終了する。

【 0 0 6 9 】

ここで、何千枚つないでもしわ寄せが起こらない画像合成の具体的な処理の一例を説明する。

ここでは、4枚の画像について考察する。

【 0 0 7 0 】

たとえば図 8 に示すように、左下が第 0 の画像 $I M 0$ 、その右が第 1 の画像 $I M 1$ 、左上が第 2 の画像 $I M 2$ 、その右が第 3 の画像 $I M 3$ とする。

第 0 の画像 $I M 0$ は動かさない。すなわち、第 0 の画像 $I M 0$ を基準画像とする。

B M（ブロックマッチング）の結果の横の成分を $bx1[0]$ 、 $bx1[1]$ 、 $bx2[0]$ 、 $bx2[1]$ とする。

縦の成分も独立に行うが、ここでは横のみを扱うことにする。

$bx1$ は左右、 $bx2$ は上下である。[]の中の数字は 0 が下または左である。

基準画像 $I M 0$ に対して右または上の画像が右または上にあるとき、B M結果は正の値となる。

今、極端な例として、一か所だけ異常な値があったときを想定し、 $bx1[0]=10$ 、 $bx1[1]=0$ 、 $bx2[0]=0$ 、 $bx2[1]=0$ とする。

【 0 0 7 1 】

ここで、1 段目の左右が 10 画素ずれていて、その他 3 箇所の境界では全くずれていないときを考える。

もし、第 0 の画像 $I M 0$ と第 1 の画像 $I M 1$ の B M 結果から第 1 の画像 $I M 1$ の位置を決め、第 1 の画像 $I M 1$ と第 3 の画像 $I M 3$ の B M 結果から第 3 の画像 $I M 3$ の位置を決める。そして、第 2 の画像 $I M 2$ と第 3 の画像 $I M 3$ の B M 結果から第 2 の画像 $I M 2$ の位置を決めていくと、第 0 の画像 $I M 0$ と第 2 の画像 $I M 2$ の位置関係に 10 画素という大きな値がしわ寄せとして生じてしまう。

【 0 0 7 2 】

本実施形態の方式では、このようなとき、10 という異常な値の影響が 2.5 ずつに分散されることを示す。なお、この処理は、一部を後述するプログラムに沿って処理される。

$xypos2()$ で隣接画像との位置関係から平行に移動すべき量を求める。

最初、第 1 の画像 $I M 1$ は - 5 画素移動するべきと計算される。

$move()$ で平行に移動する。

実際に移動させる量はその 80 %の 4 画素である。

第 0 の画像 $I M 0$ 以外の各画像 $I M 1$, $I M 2$, $I M 3$ の移動量は $pox[1]=4$ 、 $pox[2]=0$ 、 $pox[3]=0$ となる。

このことにより、B M結果bx1[0]は10から6に変化する。

それに伴い、bx2[1]は0から4に変化する。

2回目、第1の画像IM1は-1画素移動するべきと計算される。

第3の画像IM3は-2画素移動するべきと計算される。

その80%、0.8を加算するとpox[1]=4.8となる。

以下、3回目、...と計算し、32回目に、移動すべき量の総和がfxyが0.001画素を下回るため、終了となる。

このとき、平行に移動すべき画素数は7.5, 2.5, 5.0となり、各画像の位置関係も、

bx1[0]=10、bx1[1]=0、bx2[0]=0、bx2[1]=0であったものが、

bx1[0]=2.5、bx1[1]=-2.5、bx2[0]=-2.5、bx2[1]=2.5となって、誤差が分散されていることがわかる。

【0073】

以下に、ii=32, fxy=0.00083 fxyが0.001以下になったときの回数とその値を示す。

【0074】

n	ix	fx[n]	fy[n]
0	0	0.000000	0.000000
1	2	-0.000244	0.000000
2	2	-0.000244	0.000000
3	2	-0.000344	0.000000

n	pox[n]	poy[n]	fz[n]	平行に移動すべき画素数
0	0.00	0.00		
1	7.50	0.00		
2	2.50	0.00		
3	5.00	0.00		

【0075】

以下には、プログラム例の一部を示す。

【0076】

- プログラムの一部(ここから) - - - - -

```
clrpos();          // 平行に移動すべき量[pixel] pox[],fzx[],rolla[] に0を入れる。
```

```
for (ii=0;ii<1000;ii++){
    xypos2();       // 隣接画像との位置関係から平行に移動すべき量を求める。
    if (fxy<0.001){break;}
    move();         // 平行に移動する。
}
```

```
fprintf(inf,"ii=%3d, fxy=%8.5f fxyが0.001以下になったときの回数とその値\n",
ii,fxy);
```

```
xypos();           // 隣接画像との位置関係から平行に移動すべき量を求める。
move();            // 平行に移動する。
dsppos();          // 補正量を表示する。
angle();           // 補正量を角度にし、qq[n],pp[n]を更新する。
dsppos();          // 補正量を表示する。
dsperr();          // 各小画像対の誤差が1を超えるものを表示する。
step();            // qq[n],pp[n]からステップ角を作る。
```

- プログラムの一部(ここまで) - - - - -

【0077】

また、主なサブルーチンを以下に示す。

【0078】

10

20

30

40

50

- - 主なサブルーチン - - - - -

```

void xypos2(){ // 隣接画像との位置関係から平行に移動すべき量 fx[n],fy[n] を求
め、
// 平行に移動できないフラグ fz[n] を求める。( fprintf 削除 )
    int m,n,m2,n2,h,v,ix;
    double cx,cy;
    //fprintf( inf," n      ix      fx[n]      fy[n]¥n");
    fxy=0;
    for (v=0;v<ny;v++){ // すべての画像について
        for (h=0;h<nx;h++){
            m=(nx-1)*v+h; // 左右の境界
            n=nx*v+h; // 上下の境界
            ix=0;
            if ((0<skip[h][v])||((v==(ny-1)/2)&&(h==(nx-1)/2))){ // 中央の画像と確定
                済みフラグの画像は動かさない。
                fx[n]=0;fy[n]=0;fz[n]=4; // fz[n] は平行に移動できないフラグ
                if (skip[h][v]==2){fz[n]=2;} // 確定済みフラグの画像は2とする。
            }else{
                cx=0;cy=0;
                if (v!=0){ // 最下段ではないとき
                    n2=n-nx; // 真下
                    if (0<fok2[n2]){
                        ix++;
                        cx-=bx2[n2]; // 真下のものを引く
                        cy-=by2[n2];
                    }
                }
                if (v!=ny-1){ // 最上段ではないとき
                    if (0<fok2[n]){
                        ix++;
                        cx+=bx2[n]; // 自分のものをたす
                        cy+=by2[n];
                    }
                }
                if (h!=0){ // 最左端でないとき
                    m2=m-1; // 左隣り
                    if (0<fok1[m2]){
                        ix++;
                        cx-=bx1[m2]; // 左隣りのものを引く
                        cy-=by1[m2];
                    }
                }
                if (h!=nx-1){ // 最右端でないとき
                    if (0<fok1[m]){
                        ix++;
                        cx+=bx1[m]; // 自分のものをたす
                        cy+=by1[m];
                    }
                }
            }
        }
    }
    if(ix==0){

```

10

20

30

40

50

```

        fx[n]=0;fy[n]=0;fz[n]=1;
    }else{
        fx[n]=cx/ix;
        fy[n]=cy/ix;
        fz[n]=0;
    }
    fxy+=fabs(fx[n])+fabs(fy[n]);
}
}
}
// *****
void move(){          // 平行に移動する。
    int m,n,h,v;
    for (v=0;v<ny;v++){          // 画像の中心位置 (画素)
        for (h=0;h<nx;h++){
            n=nx*v+h;
            if (fz[n]==0){        // 周囲から隔絶されていないとき
                pox[n]+=-fx[n]*0.8;
                poy[n]+=-fy[n]*0.8;
            }
        }
    }
    for (v=0;v<ny;v++){          // 左右の位置関係
        for (h=0;h<nx-1;h++){
            m=nx*v+h;
            n=(nx-1)*v+h;
            bx1[n]+=-(fx[m]-fx[m+1])*0.8;
            by1[n]+=-(fy[m]-fy[m+1])*0.8;
        }
    }
    for (v=0;v<ny-1;v++){        // 上下の位置関係
        for (h=0;h<nx;h++){
            n=nx*v+h;
            bx2[n]+=-(fx[n]-fx[n+nx])*0.8;
            by2[n]+=-(fy[n]-fy[n+nx])*0.8;
        }
    }
}
// *****

```

【 0 0 7 9 】

以上説明したように、本実施形態の第1の構成によれば、CMOSイメージセンサを搭載したデジタルカメラを用いても解像度を低下させずに高速フレームレートで撮影し、パララックスの影響、周辺減光の影響、レンズ歪の影響を軽減することができる。しかも、高品質のパノラマ画像を作成することができる。

合成する画像の数にかかわらず、正確な合成を行うことが可能で、また、色ムラの発生を抑止することができる。

そして、レンズ歪補正係数を実写画像から抽出することができたため、厄介なキャリブレーションの作業が不要になり、精度も格段に向上する。

何千枚つないでもしわ寄せが出ない方法により、撮影枚数を気にすることなく、必要な

10

20

30

40

50

範囲を必要な解像度で撮影することができるようになる。

【 0 0 8 0 】

次に、第 2 の構成について説明する。

【 0 0 8 1 】

[第 2 の構成]

連続撮影した画像の空間位置記録について説明する。

【 0 0 8 2 】

< 概要 >

連続写真によるパノラマ撮影は、空間を分割して 1 枚の写真に収める作業となる。その写真からパノラマを作成する際に、撮影時の空間情報を用いて逆の演算を行うことで高精度なパノラマ写真とすることができる。

10

本実施形態では、写真を撮影する際にセンサと画像から撮影した空間の情報を計算し、その情報を写真 1 枚 1 枚に付与しておく事でパノラマ写真生成に役立てる。

【 0 0 8 3 】

< 撮影した空間の情報を付与する >

たとえば、パノラマ撮影をする場合、一点に固定してモータでレンズを駆動させ撮影方向を変化させる。

この条件で撮影した写真は、撮影方向が異なるだけでカメラ装置 10 の位置つまり焦点位置は固定となる。よって、ここではとある一点から固定画角でその周囲を撮影した画像に限定する。

20

このような撮影方法の場合、撮影した空間の情報は以下の二つとなる。

すなわち、どこを狙って撮影したか(視線ベクトル)の情報と、視線ベクトルを軸とした回転角(ロール)の情報である。

【 0 0 8 4 】

< 投影球と空間の定義 >

空間を撮影した写真は 1 枚の面に投影される。

空間をパノラマ撮影すると考えたとき、全方位を扱うには撮影者の周辺に球が存在しそこに投影されると考えると画像処理が行い易い。この球を用いて視線ベクトルを定義することで座標空間も決定される。

カメラ装置 10 が存在する焦点位置を原点(0,0,0)とし、投影球を半径 1 と定義する。

30

水平時の正面方向は Z 軸の 1、 $f(0,0,1)$ とすると視線ベクトルは(0,0,0)から $f(0,0,1)$ へ向かうベクトルとなる。

視線ベクトルは長さ 1 の単位ベクトルとなり、どこを向いても長さは 1 である。

視線ベクトル v_1 だけではフレームのロールが記録できないため、もう 1 つのロールベクトル v_2 を記録する。これは画像の上方向を示す情報で(v_2-v_1)が画像の上方向を示すベクトルとなる。

これにより、画像の撮影方向を 2 つのベクトル(投影球上の 2 点)で示すことができ、全方位で密度の差無く指し示すことができる。

【 0 0 8 5 】

< 相対移動と絶対座標 >

40

撮影時の空間情報には相対情報と絶対情報の二種類が存在する。

パノラマ画像を作る目的においては、どこを向いて撮影した画像かという絶対的位置情報があれば良いのだが、確実な絶対情報が得られないため相対情報から積算したり、大雑把な絶対情報を使って補正したりすることになる。

レンズ駆動型パノラマカメラにおいては、レンズを移動するシナリオは絶対情報だが、撮影時の揺れ、レンズ駆動時の誤差混入、姿勢感知センサの精度の粗さ等が相対情報として加算されるため精密な絶対値は計算で求めることになる。

【 0 0 8 6 】

< 相対移動の空間展開 >

今、画像認識と姿勢センサにより正確な相対情報が入手できたとする。

50

現在の画像フレーム f_1 が 1 つ前の画像フレーム f_2 より、位置的に (dx, dy) 移動しており、フレームロールが r_z だけロールしていた場合、画角から x 軸 y 軸の回転量が r_x, r_y と計算できる。このとき f_1 の視線ベクトル v_1 は f_2 の視線ベクトル v_2 が (r_x, r_y, r_z) 回転したことになる。

この情報を元に投影球上の絶対位置を求めていくのであるが、 v_2 の位置から回転 (r_x, r_y, r_z) と求めていくのは少々計算が複雑となる。

そこで、最新の画像 f_1 を真正面 $v_1(0,0,1)$ と固定してしまい、 f_2 以降投影球上に配置された画像を球ごと $(-r_x, -r_y, -r_z)$ だけ回転させてしまう。つまり、最新の画像 f_1 を基準に、他を相対的に移動させる。

これを繰り返していくと、最後の画像の位置が $(0,0,1)$ となるがそれ以外の全ての画像の絶対座標が求まっている。

ロールの情報もベクトルとして、視線とロール二つのベクトルにしたのはこの相対的回転(球の方を回転させること)が容易になるからである。

【0087】

< 相対移動値の積算と絶対情報とのズレ >

画像の空間配置のために相対情報だけを用いたが、実際には姿勢センサ 14 等により絶対的なロールや上下方向の傾き情報も入手できている。ただ、姿勢センサ 14 により得られる絶対情報はパノラマ写真の生成に必要な精度と比べ荒く、その値をそのまま使うのは困難である。

一方、相対情報は画像認識から得るため高い精度を持つが、それでも誤差は含まれる。相対情報で繋いでいくと積算誤差で若干の誤差が大きくなずれとなって現れる。

そこで、姿勢センサ 14 による絶対情報をこの積算誤差が生じていないかの確認として用いる。

相対移動の空間展開処理の途中で、ある程度の間を空けつつ姿勢センサの絶対値と比較をしていく。あまりにも相対移動が姿勢センサの絶対値と離れてきていたら姿勢センサの絶対値を用いて補正を行う。そしてまたその位置から相対的に移動の積算を行っていく。

【0088】

図 15 は、連続撮影した画像とセンサ情報を基に空間配置する方法を示す機能ブロック図である。

【0089】

図 15 において、角速度センサ 142 の検出信号に対して機能ブロック 41 でゼロ基準値を設定して、移動量積分部 42 で移動量を積分する。

また、撮像素子 12 を通して撮影したフレーム間の画像比較と移動量を検出部 43 で検出する。

移動量積分部 42 と検出部 43 の出力から協調補正ロジック 44 で協調補正し、相対位置積分部 45 で相対位置を積分し、絶対位置情報を得る。

そして、その絶対位置情報を加速度センサ 141 の検出結果に基づいて絶対位置補正部 46 で補正し、配置部 47 でフレームの空間位置を決定し、配置する。

【0090】

< 空間座標情報とパノラマ写真 >

以上のような計算を撮影しながら行っていく、画像とともに撮影方向といった空間座標情報をメタデータとして同時に記録する。

このメタデータだけでもパノラマ写真を生成することができるが、後処理より精密な調整とオーサリングを行う際の基本データとしても用いることができる。

【0091】

これまでは空間における撮影方向というメタデータが存在しなかったためパノラマ写真を精密化することができなかったが、本実施形態では、それを解決するために撮影時に座標情報を付与する。

【0092】

10

20

30

40

50

このように、第2の構成では、連続した画像を、画像認識技術で得られたフレームの移動情報と、姿勢センサからの移動情報を用いて空間的に配置する。画像認識では情報を得られない部分を姿勢センサ情報で補い、姿勢センサ情報は画像認識の成功確認や失敗時の補助座標として用いる。空間的に配置された画像は1つのパノラマ画像として完成する。

この方法を使うと正面付近のみならず、真上も真後ろも正しく表現でき、全方位、あるいは全天球撮影に対応させることができるようになる。

正面付近のみの場合でもより広角画面でも誤差なく仕上げる事が可能になる。

無論、手持ち撮影に十分に対応することができ、高精細な画像を得ることができる。

【0093】

次に、第3の構成について説明する。

10

【0094】

[第3の構成]

連続撮影した画像による位置認識の補正について説明する。

【0095】

<概要>

連続撮影した写真に高精度な撮影位置情報を付与するために、本構成では、姿勢センサと画像認識を合わせた手法、いわゆる「ダイナミックキャリブレーション」を用いる。

【0096】

<連続撮影と撮影位置の情報>

連続撮影した写真を用いてパノラマ写真を合成する際、画に高周波成分が含まれず画から連続性を特定できない場合がある。

20

そのような場面では連続するフレームがどれくらい移動しているのかといった情報を得ることができず、パノラマ写真全体が生成できないことになる。

そういった場面においても位置情報を得られるよう、姿勢センサ14を併用し撮影中に移動情報、位置情報を写真と同時に記録する。

【0097】

<姿勢センサの利用>

姿勢センサ14は3軸の角速度センサ142と同じく3軸の加速度センサ141を同時並列に用いる。

角速度センサ142で今どれくらいの速度でカメラを回転させているかをとらえ、加速度センサ141で水平方向の傾きを感知する。

30

できるだけ撮影した写真で移動情報を得ていくが、画の都合で画像認識ができないような場面では前フレームからの移動量を姿勢センサ14から得る。

全体としての姿勢センサ14の検出結果の変化量と、画像認識による移動量を比較し相互に影響することでより厳密な位置情報を得ることができる。

【0098】

<姿勢センサの問題点>

目的とするパノラマ写真撮影においては、姿勢センサ14の精度よりも画像の精度の方が高くなっている。姿勢センサ14の情報のみで位置情報としてしまうと、パノラマ写真として成り立たないほどの粗さとなってしまう。

40

精度としては期待せずに、本当に情報が取れないときの補佐とする様な使い方となる。

一般的な物理センサと同じように、姿勢センサも安定した値を出さず常に揺れている。

また、制止時のゼロ位置が状況によって変化するため、撮影前に静止状態を作り出しゼロ位置の値を計測する必要がある。その値の計測後はゼロ値からの変位でどれくらい移動したかを計測する。

【0099】

<画像認識とその補正>

本構成では、連続したパノラマ写真を撮影しながら、姿勢センサの情報もメタデータとして記録していく。

この方法では姿勢センサ14の揺れが大きすぎるため、後々パノラマ合成時にメタデー

50

タの情報が使いにくいという難点がある。

そこで、撮影時に画像認識によるメタデータを補正計算しながら記録していく。

【0100】

こうしたメタデータを記録するにはカメラが今どこを向いているのかといった空間情報を内部に保持し、それを更新していくのであるが、様々な要因でこの値の精度が落ちていく。

そこで、本実施形態では、画像認識と姿勢センサからこの内部保持している空間情報をリアルタイムに補正更新していこうというのが、ダイナミックキャリブレーションである。

【0101】

連続したパノラマ写真を撮影する際、モータ駆動により予め移動シナリオが存在する場合と、手でスワイプ移動させるため移動シナリオが存在しない場合の二通りが存在する。

モータ駆動で移動シナリオが存在する場合は、予めおおよその撮影位置が分かっているが撮影中の揺れや移動までは考慮できない。そうした、撮影時の変化を検知するために姿勢センサ14を用いる。

姿勢センサ14で撮影時の変化を感知したら画像認識で実際の移動シナリオとどれだけずれたかを精査する。この際の精査基準に姿勢センサ14の移動量を用いると画像認識も容易となる。

こうして、期待する移動シナリオとの誤差が算出できたら、移動シナリオの値に加算し実際に撮影した位置情報を撮影写真のメタデータとして記録していく。

手作業によるスワイプ撮影の場合は移動シナリオが存在しないため、毎フレームの撮影毎に前フレームと今フレームを画像認識による比較で移動量を算出する。

このときどれだけ動いたかが分かりにくいので、姿勢センサ14の情報からおおよその移動量を得てその値を元に画像認識を行うことで精度の良い移動量算出が行える。

もし画像認識が困難だった場合は姿勢センサから得た移動量を記録しておき後ほど前後のフレームの位置関係などと照らし合わせて座標を決定することになる。

【0102】

図16は、連続撮影した画像とセンサ情報を対応させて高精度化する方法であって、静止時のセンサ値のゼロ補正を示す機能ブロック図である。

【0103】

図16において、撮像素子12を通して撮影したフレーム間の画像比較と移動量を検出部51で検出する。

角速度センサ142の検出信号、加速度センサ141の検出信号、および検出部51の検出信号を基に静止検出部52で静止検出を行い、静止時の角速度センサの基準値を得る。

そして、記録部53により基準値の決定を行い、その基準値をメモリ17に記録する。

【0104】

図17は、連続撮影した画像とセンサ情報を対応させて高精度化する方法であって、移動情報の協調により高精度化を示す機能ブロック図である。

【0105】

図17において、角速度センサ142の検出信号に対して機能ブロック54でゼロ基準値を設定して、移動量積分部55で移動量を積分する。

また、撮像素子12を通して撮影したフレーム間の画像比較と移動量を検出部51で検出する。

移動量積分部55と検出部51の出力から協調補正ロジック56で協調補正し、高精度の相対移動情報を得る。

【0106】

このように、第3の構成では、連続した画像の記録において、画像認識技術で得られたフレームの移動情報と、姿勢センサからの移動情報を対応させる。片方だけでは不明な、

10

20

30

40

50

画像のピクセル画角、姿勢センサの静止時の値、姿勢センサ値のピクセル画角対応といった情報を算出する。

したがって、画像から認識する方法も姿勢センサで検出する方法も単独では満足な精度のものが得られなかったが、その両者を協調させるというこの技術により、精度と安定度を格段に向上させることが可能となる。

【 0 1 0 7 】

ここで、上述した第 2 および第 3 の構成について、図 1 8 ~ 図 2 1 に関連付けてさらに詳述する。

【 0 1 0 8 】

< 撮影写真と撮影者の回転移動 >

10

パノラマ写真を撮影する際、撮影位置が変動すると視差による非連続性が発生する。

視差による非連続性は撮影後の画像処理で補正することができない。

このため、パノラマ写真を撮影する場合、撮影者とカメラは特定の位置で静止しカメラの焦点が一箇所になるようその場で回転しながら写真を撮影する。

このとき、撮影後の異なる 2 枚の写真内で見た目の移動距離は撮影時の回転量に比例する。

写真がデジタル画像であり、そのサイズがピクセル数で表現できる場合、2 枚の画像間で何ピクセルの移動であったかは撮影時どれくらいの回転移動距離があったか逆算することが可能であるが、このとき必要なパラメータとして画角がある。

画角とは、写真一枚に写っている左右幅もしくは上下幅が撮影空間で何度の範囲かを示す数値である。

20

画角は撮影前に計測し与えられるパラメータであり、撮影中は揺るがないものとする。

水平画角が 3 0 度であり、撮影したデジタル写真の横方向のピクセル数が 1 0 0 0 ピクセルであった場合、1 ピクセルあたりの撮影空間角は 0 . 0 3 度となる。つまり、2 枚の写真間で 8 0 0 ピクセル移動していることが認識できたとき、実際の撮影は 2 4 度回転して撮影されたものの計算される。

この 1 ピクセルあたりの画角が最も重要な初期値として適用される。

【 0 1 0 9 】

1 ピクセルあたりの画角 = フレーム画角 / フレームピクセル数

写真 2 枚の撮影時回転量 = 写真 2 枚のピクセル上での移動量 * 1 ピクセルあたりの画角

30

【 0 1 1 0 】

実際の 1 ピクセルあたりの画角は初期値としてあらかじめ計測して保持しておく。

【 0 1 1 1 】

< 角速度センサと回転量 >

角速度センサは現在の角速度を出力する。

時間経過によって出力値が変動するため、角速度の変化を知ることができるが、その値が直接回転量を表してはいない。角速度センサから回転角を得るためには、積分値の単位を規定することが必要となる。

角速度センサを一定時間ごとに計測を行い、この計測間隔を重要なパラメータとして固定する。

40

計測した角速度を時間で積分し、その間に実際に回転した量を外部からの計測で知る必要がある。積分された角速度を実際に回転した量で除算し、1 度あたりの各速度積分値を求める。

以降は、角速度積分値を 1 度あたりの角速度積分値で除算することで、比例的に回転移動量を求めることができる。

実際の 1 度あたりの角速度積分値は初期値としてあらかじめ計測して保持しておく。

【 0 1 1 2 】

< ダイナミックキャリブレーション >

角速度センサの出力は相対的な角速度であるが、すぐれた角速度センサでない限りその出力は環境によって変化する。その変化が実際の計測に影響を及ぼすので、計測の度に補

50

正を行う必要がある。

パノラマ撮影に特化して、その補正を撮影したパノラマ写真からのフィードバックで自動的に行う処理を、ここではダイナミックキャリブレーションという。

角速度センサで、環境によって変動するには静止時零点位置と1度あたりの角速度積分値の二つである。それと相対移動により積分誤差の積み上げがあり、以上3項目の補正を行う。

【0113】

<角速度センサの零点補正>

図18は、角速度センサの零点補正処理のフローチャートである。

角速度センサ142の零点補正においては、図18のステップST1～ST16の処理が行われる。

角速度センサ142が静止しているときの出力値が正確にわからないと角速度を検出することができない。ところが、この静止時の零点が温度等の環境によって変化してしまうことがある。

この零点ドリフトを画像マッチングの結果から補正していき、撮影時の正確な零点を求める。

スタート時の角速度センサ142の零点出力値は初期値としてあらかじめ設定されている値を用いる。

2フレーム間の画像マッチングを行い(ST1～ST3)、マッチングが高周波成分を含んだたしかなものであり、かつ、X、Y、Z軸方向に移動がなかった場合、X、Y、Z軸方向の角度センサは零点を指しているものとして数値をサンプリングする。

このとき零点としてサンプリングされた値を用いて補正を行う(ST4～ST15)。

なお、軸方向に移動が検出された場合は零点ではないためサンプリングを行わず、零点補正は行われない。

サンプルを行ったらサンプル数をインクリメントし、零点値を補正する。

補正は現在の零点値とサンプル値の差分をサンプル数で除算したものを加算し、平均値の算出とする。

【0114】

補正零点値 = 零点値 + (サンプル値 - 零点値) / サンプル数

【0115】

<角速度センサの移動量補正>

図19は、角速度センサの移動量補正のフローチャートである。

角速度センサの移動量補正においては、図19のステップST21～ST26の処理が行われる。

1度あたりの角速度積分値は、角速度センサの角速度積分値から回転角を求めるパラメータであるが温度等の環境によって変化してしまうことがある。

この1度あたりの角速度積分値を画像マッチングを行い(ST21～ST23)、マッチングの結果から補正、更新していき、撮影中の正確な値を求める(ST24～ST26)。

2フレーム間の画像マッチングを行い、マッチングが高周波成分を含んだ確かなものであった場合、画像マッチングによって得られたX軸、Y軸、Z軸それぞれの移動量とそのときの角速度積分値から1度あたりの角速度積分値を求める。

【0116】

1度あたりの角速度積分値 = 角速度積分値 / (1ピクセルあたりの画角 * X軸ピクセル移動量)

【0117】

補正1度あたりの角速度積分値 = 1度あたりの角速度積分値 + (サンプル値 - 1度あたりの角速度積分値) / サンプル数

【0118】

<加速度センサによる角速度センサの補助>

角速度センサは相対的な角移動量を出力する。

現在どの位置にいるかという絶対位置情報はそれまでの相対値を積分することで算出される。

その相対値にわずかなずれやノイズが含まれていた場合、積分する時間が長引くほど大きなずれを生じる可能性がある。

角速度センサは重力加速度の検出でY軸方向の回転（チルト）とZ軸方向の回転（ロール）の絶対値を得ることができるが、パノラマ撮影に対して大きな単位でしか検出できず有用度では角速度センサにおとる。

しかし、絶対値が出てくるというメリットから、定期的に相対移動距離の積分値と比較して積分値の補正、絶対値の補正を行うことができる。

加速度センサで十分検出できるくらいの絶対量を移動したら、そのタイミングで相対移動距離の積分値による絶対位置と比較して必要であれば補正を行う。

【 0 1 1 9 】

< 画像から得る移動とセンサから移動 >

図 2 0 は、移動量の取得方法のフローチャートである。

移動量の取得方法においては、図 2 0 のステップ S T 3 1 ~ S T 3 5 の処理が行われる。

角速度センサ 1 4 2 の解像度と、画像マッチングによる移動量の解像度では画像マッチングの方が遥かに精度の高い数値となる。このため、相対移動距離はできるだけ画像マッチングから算出できるものを用いる（S T 3 3 , S T 3 4 ）。

一面が同色の空ならの高周波成分が含まれない画像間ではマッチングが取れないが、そのときは角速度センサ 1 4 2 の数値を用いて相対移動量を算出していく（S T 3 3 , S T 3 5 ）。

【 0 1 2 0 】

< 撮影した写真から空間座標を割り当てる方法 >

図 2 1 は、撮影した写真から空間座標を割り当てる方法のフローチャートである。

撮影した写真から空間座標を割り当てる方法においては、図 2 1 のステップ S T 4 1 ~ S T 4 7 の処理が行われる。

上記のように撮影したパノラマ写真すべてに、画像マッチングと角度センサから得た前フレームからの相対的回転移動量を求めることができる（S T 4 1 ~ S T 4 3 ）。

パノラマ写真を構成するためには、これら相対的回転移動量から絶対空間座標への割り当てを行う必要がある。

撮影した写真はすべて同じ画角なので、撮影した中央の点、つまりカメラが向いていたベクトルのみに注視して割り当てを考慮することができる。

前フレームとの相対的回転移動量というのは、このカメラが向いていた方向、撮影視線ベクトルが前フレームのベクトルとどれだけの角をなしているかと表現することもできる。

配置にあたって視線ベクトルだけに簡略化するとフレームZ軸方向の回転、つまりロールが取れなくなる。

そこで、フレームの真上Y軸上にずらしたベクトルをもう一つ用意することでフレームのロールを表現することにする。

この2つのベクトルがカメラの撮影方向とZ軸のロールを表し、回転処理を行ってもフレームの情報を保持し続ける。

空間に配置する際、新しいフレームは必ず空間正面位置 $a(0, 0, 1.0)$ には位置する。

相対的回転移動量が出ている場合は、その分だけ逆方向に前のフレームすべてを回転させてから空間正面位置 $a(0, 0, 1.0)$ を配置する（S T 4 4 ~ S T 4 6 ）。

前のフレームから移動ではなく、現在のフレームから過去のフレーム画どれだけ移動したかということを基準にしてあつかう。

今回撮影したフレームをAとし、前回撮影したフレームをBとする。

フレーム A を基準にして、フレーム B がどれだけ相対的回転移動したかを算出する (S T 4 3)。

現在の A のカメラ位置からみて B のカメラ位置が X 軸方向に r_x 、Y 軸方向に r_y 、Z 軸方向に r_z だけ回転移動したものと算出されたら A 以外過去に撮影されたフレームの撮影方向ベクトルおよびロール指標ベクトルすべてに、 r_x 、 r_y 、 r_z の回転をかける。

回転マトリクスは一般的な三次元空間のものでよい。

【0 1 2 1】

X 軸方向の回転

$$\begin{aligned} x_2 &= x * \cos(r_x) - y * \sin(r_x) \\ y_2 &= y * \sin(r_x) + z * \cos(r_x) \\ z_2 &= z \end{aligned}$$

10

【0 1 2 2】

Y 軸方向の回転

$$\begin{aligned} x_2 &= x * \cos(r_y) - z * \sin(r_y) \\ y_2 &= y \\ z_2 &= x * \sin(r_y) + z * \cos(r_y) \end{aligned}$$

【0 1 2 3】

Y 軸方向の回転

$$\begin{aligned} x_2 &= x \\ y_2 &= y * \cos(r_z) - z * \sin(r_z) \\ z_2 &= y * \sin(r_z) + z * \cos(r_z) \end{aligned}$$

20

【0 1 2 4】

そのようにして全体を回転させつつ、正面という定位置に新しいフレームを張っていくことで相対的回転移動量から絶対空間座標への転写が行える。

最終的にすべてを適用し終えたとき、すべてのフレームは適した絶対座標を持ちうる。

ただし、最後のフレームが基準になっているので、任意のフレームを基準に相対移動する必要がある場合もある。

【0 1 2 5】

次に、第 4 の構成について説明する。

30

【0 1 2 6】

[第 4 の構成]

第 4 の構成では、視差による影響や移動体による影響を検出した場合に、表示装置 1 8 や発音部 2 0 により警告を出して撮影のやり直しを促す。

また、第 4 の構成では、移動体の検出に関し、重複率を 5 0 % 以上として被写体のどの部分も少なくとも 2 枚の画像にまたがって写るようにし、隣接画像間の移動ベクトルの類似性で視差による影響や動被写体を検出する。

【0 1 2 7】

広い範囲の被写体をさっと一振りで複数の短冊状画像を撮影して 1 枚に合成するカメラ装置 1 0 において、近距離の被写体がパララックスの影響をどの程度受けているかを検出し、カメラの視点を中心として撮影し直すことを促す。

40

一般に広角のカメラの視点はレンズの直後にあり、カメラを手にとって手首を中心に回すのが理想的である。

こうしてカメラの視点を中心として撮影したものは近距離の被写体が含まれていても正しく合成することができる。

本実施形態のカメラ装置 1 0 は、複数の画像をすべて短冊状に撮影するのでカメラの視点から少々ずれた位置を中心として撮影してもその影響が出にくいという長所も持っている。

しかし、それでもカメラを手にとって肩を中心にして回して撮影した場合、カメラの視点よりかなり後方を中心として回すことになり、パララックスの影響を強く受ける。

50

すべてが遠方の景色なら殆ど問題ないが、近距離の被写体が含まれていると隣接画像との位置関係が遠距離の被写体と異なり、正しく合成することができない。

そこで、第4の構成では、パララックスの影響で正しい画像合成ができないことが検出された場合に、視点を中心にして回すように指示を出して撮り直しを促すことにしている。

【0128】

[視差検出の方法]

視差の検出方法について説明する。

時間的に隣接する2枚の画像の重複する領域内で複数のブロックマッチング(BM)を行い、その結果の移動ベクトルを求める。

10

通常、正しくスイープされているとBM結果がほぼ同一のベクトルになる。

肩中心に回したとき、遠距離の景色の一部に近距離の被写体があると、異なるベクトルの値になる。

近距離の被写体と遠距離の被写体の境目では画像の変化が激しいため、BM結果は正しく得られない。この方法で視差を検出する。

【0129】

以下に、視差検出の具体的な処理例について示す。

以下の処理は、システムコントローラ16の画像信号処理部161とマイクロコンピュータ162により協働して行われる。

【0130】

<視差検出の方法>

[ラフ合成]

カメラ装置10を左から右に向かって回転移動させ、120度程度の範囲を数10枚撮影する。

隣接する画像には同一の被写体が写っている領域(重複領域)が十分に存在する。

撮影時のカメラ装置10の動きが姿勢センサ14によって検出され、細かい時間間隔で記録されている。

このデータは撮影された画像と同期を取りながら記録されているので、精度はさほど高くないが、各画像がどの方向を向いて撮影されたのかを知ることができる。

この情報に基づいて各画像を経度緯度平面上に配置する。

30

この状態では隣接画像の重複領域は、100画素程度の画素はあるものの、ほぼ正しい位置に置かれている。

ここから精密自動合成のルーチンに入る。

【0131】

[精密自動合成]

重複領域内で複数の場所で動き検出(ME;動き探索)を行う。

ME、すなわち動き検出では、FFTによる位相限定相関法を用いている。これは、特徴点抽出法やその他の方法であっても適用可能である。

平行移動のみの場合は、MEは一つで良い。

MEが2個あれば互いの傾斜がわかる。

40

3個あれば、レンズ歪みの係数がわかる。

重複領域内に動くものがなく、遠景で手持ちでもその影響が生じない場合はMEの数は少なくても良い。

しかし、動くものがあったり、近距離で視差の影響があったりすると、MEの数が少ないと何が起きているかを把握することができない。

そこで、重複領域内で可能な限り、多くのMEを行う。

多くのMEの結果の移動ベクトルがほぼ同じ値の場合、一方の画像を平行移動によって合わせこむことができる。

ほぼ同じ値でない場合でも、上部から下部にかけて一様に変化する場合、一方の画像を傾斜させることで合わせこむことができる。

50

しかし、異なるME結果が重複領域内に存在する場合、合成ができない。

その理由は、動く被写体が存在するか、近距離と遠距離の被写体が混在するときに視点を移動させながら撮影したかである。

【0132】

[MEの方法]

最初はMEを行う画像を縮小して大まかに行う。

だんだん縮小率を下げ、最後は等倍の画像で行う。

MEのブロックサイズを変えたり、ブロックの中心間距離を狭くして、より詳細な移動ベクトルを取得できるようにする。

【0133】

[ME結果の評価]

多くのMEの結果から正しいスイッチングが可能か否かを判断し、不可能な場合、その旨を表示して撮影のやり直しを促す。

可能な場合はそのままスイッチング処理を行い、合成結果を表示すると共に、記録媒体(メモリ)に記録する。

【0134】

[移動被写体の振る舞い]

次に、移動被写体の振る舞いについて説明する。

時間的に隣接する2枚の画像の重複する領域内で複数のブロックマッチング(BM)を行い、その結果の移動ベクトルを求める。

移動の方向に応じたベクトルになり、静止部分との分離が可能である。

移動物体と静止被写体の境目では画像の変化が激しいため、BM結果は正しく得られない。横にスワイプしている場合、近距離の静止被写体による視差なのか、横に移動する被写体なのかを識別することができない。

そこで、視差と動被写体を区別しないで警告を発するようにする。

これを検出した場合、現在の技術ではこれを違和感なくステッチングすることができない。

そこで、本実施形態においては、「単に撮りなおし」、または「撮り方を変えて撮りなおし」を促すように警告を発する。

たとえば、「視差または移動体の影響が出ています。回転の半径を小さくして撮りなおしてください。」との警告を発する。

【0135】

このように、第4の構成では、動被写体の存在が撮影直後にわかるため、撮りなおしができるようになる。

その結果、視差の影響が撮影直後にわかるため、撮りなおしができるようになる利点がある。

【0136】

次に、第5の構成について説明する。

【0137】

[第5の構成]

第5の構成では、スワイプ角速度(ユーザがカメラを振り回す速さ)の適正値を知らせ、速すぎる場合に警告を出して撮影のやり直しを促す。

前述したように、マイクロコンピュータ162は、表示装置18、たとえばLCDの画面に横軸に時間、縦軸に姿勢センサ(ジャイロセンサ)の出力(スワイプ角速度)をグラフにして表示する。

水平画角、水平画素数、シャッタ速度が設定されると最大スワイプ角速度が決まるので、その60%~80%を適正範囲RNGとして、図4に示すようにグラフに表示する。

操作手順の概略は以下の通りである。

[1]操作部19のスタートボタンを押してカメラを回し、その後スタートボタンを離す。

[2]スタートボタンが押されている間のスワイプ角速度が、図4に示すように、表示装置

10

20

30

40

50

18の画面に表示される。

[3]適正範囲RNGより遅い場合には警告はないが、一瞬でも範囲を越えて速い場合には警告音が出る。

【0138】

このように、第5の構成では、適正な速さを教えてくれるので、速すぎて重なりがなくなったり、遅すぎて狭い範囲しか撮れなかったりという不都合がなくなった。

【0139】

ここで、スイープ速度の計算処理例について、図22に関連付けて説明する。

【0140】

<スイープ速度の計算>

露光時間、画素数、1ライン読み出し時間、画角、重複率、フレームレート等が決まったとき、ぶれの角度、ぶれの画素数、その他の問題が生じないためのスイープ速度を計算で求める方法について説明する。

3つの式で得られるスイープ速度の最も遅い値がその条件における最高角速度である。

【0141】

図22に示す表は、画角やスイープ速度など、各種パラメータを与えたとき、ぶれの画素数やフレームレートがどんな値になるかを計算したものである。

[1]～[6]は条件を変えた場合である。

以下に、図22の条件[1]を例にとって説明する。

【0142】

ぶれの角度 $a b 2$ とぶれの画素数 $n b 2$ とフレームレート f は、スイープ速度 $v p$ 、画角 $t h$ 、水平画素数 H 、重複率 k を用いて、図22の表の右端の計算式から次のようになる。

【0143】

$$\begin{aligned} a b 2 &= v p \cdot (t s + n \cdot r s) \cdot 1000 \\ n b 2 &= v p \cdot (t s + n \cdot r s) \cdot H / t h \\ f &= 1000 / (100 - k) \cdot H \cdot v p / n / t h \end{aligned}$$

【0144】

これらより、スイープ速度 $v p$ は次のようになる。

【0145】

$$\begin{aligned} v p &= 1000 \cdot a b 2 / (t s + n \cdot r s) [deg] \cdots (1) \\ v p &= n b 2 \cdot t h / H / (t s + n \cdot r s) [deg/sec] \cdots (2) \\ v p &= (1000 - k) / 100 \cdot n \cdot t h \cdot f / H \cdots (3) \end{aligned}$$

【0146】

ここで、ぶれの角度 $a b 2 = 0, 28$ 度、露光時間 $1 [msec]$ 、短辺の画素数 $n = 400$ 画素、1ライン読み出し時間 $r s = 7.8 [\mu sec]$ とすれば、スイープ速度は $v p = 68 [deg/sec]$ となる。

また、ぶれの画素数 $n b 2 = 19.9$ 画素、長辺の画角 $t h = 50$ 度、水平画素数 $H = 3560$ 画素とすれば、スイープ速度は $v p = 68 [deg/sec]$ となる。

また、重複率 $k = 20\%$ 、フレームレート $f = 15.13$ のときスイープ速度は $v p = 68 [deg/sec]$ となる。

【0147】

したがって、上記(1)、(2)、(3)式の右辺のパラメータが変換したときスイープ速度はこの式によって制限を受ける。

式(1)で得られた $v p$ の値より速くスイープすると光学手振れ補正素子の動作限界を超える。

式(2)で得られた $v p$ の値より速くスイープするとぶれの量が許容値を超える。

式(3)で得られた $v p$ の値より速くスイープすると重複量が減り、場合によっては重複しなくなる。

【0148】

10

20

30

40

50

以上詳細に説明した第 1 ~ 第 5 の構成は、それぞれ単独であるいは全てをカメラ装置 10 に採用することも可能であり、また、これらを適宜組み合わせ採用することも可能である。

【 0 1 4 9 】

なお、以上詳細に説明した方法は、上記手順に応じたプログラムとして形成し、CPU 等のコンピュータで実行するように構成することも可能である。

また、このようなプログラムは、半導体メモリ、磁気ディスク、光ディスク、フロッピー（登録商標）ディスク等の記録媒体、この記録媒体をセットしたコンピュータによりアクセスし上記プログラムを実行するように構成可能である。

【図面の簡単な説明】

10

【 0 1 5 0 】

【図 1】本発明の実施形態に係る画像処理装置を採用したカメラ装置の構成例を示すブロック図である。

【図 2】本実施形態に係るカメラ装置において広角撮影を行った場合を概念的に示す図である。

【図 3】精密合成処理部のブロック図である。

【図 4】姿勢センサの出力（スイープ角速度）をグラフにして表示する場合を示す図である。

【図 5】本実施形態の第 1 の構成における撮影形態を説明するための図である。

【図 6】CMOS イメージセンサの露光時間と蓄積電荷の読み出し時間、そして光軸制御時間の関係を示す図である。

20

【図 7】相互電力スペクトル(CPS)を用いたトランスレーションにおけるステッチング画像を示す図である。

【図 8】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、条件の 4 枚の画像を選択する処理を示す図である。

【図 9】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、一つの境界で 3 箇所 BM を行う例を示す図である。

【図 10】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、レンズ歪があると BM の結果が弓なりになることを示す図である。

【図 11】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、チルト角が正しくないと左右方向の誤差が生じる例を示す図である。

30

【図 12】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、左右の境界で上下の伸縮が起きているときに横にずれができる例を示す図である。

【図 13】ブロックマッチング(BM)でパラメータを抽出する処理を説明するための図であって、画像が回転する場合に生じる誤差の例を示す図である。

【図 14】ブロックマッチング(BM)でパラメータを抽出する処理後に多くの枚数に拡張して平行移動を行って最も誤差が小さくなるようにする処理を説明するための図である。

【図 15】連続撮影した画像とセンサ情報を基に空間配置する方法を示す機能ブロック図である。

40

【図 16】連続撮影した画像とセンサ情報に対応させて高精度化する方法であって、静止時のセンサ値のゼロ補正を示す機能ブロック図である。

【図 17】連続撮影した画像とセンサ情報に対応させて高精度化する方法であって、移動情報の協調により高精度化を示す機能ブロック図である。

【図 18】角度センサの零点補正処理のフローチャートである。

【図 19】角速度センサの移動量補正のフローチャートである。

【図 20】移動量の取得方法のフローチャートである。

【図 21】撮影した写真から空間座標を割り当てる方法のフローチャートである。

【図 22】スイープ速度の計算処理例について説明するための図である。

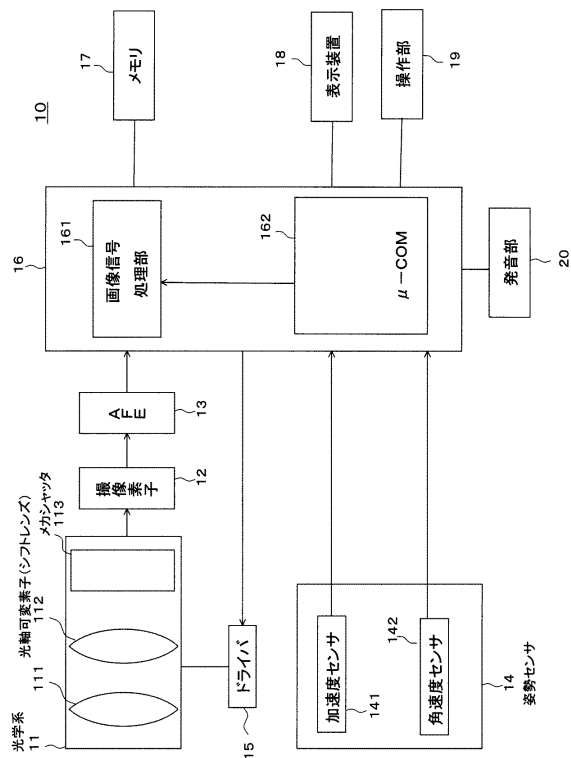
50

【符号の説明】

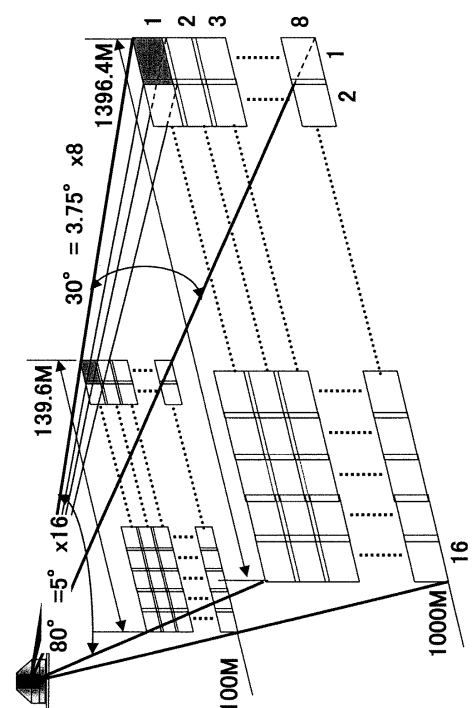
【 0 1 5 1 】

1 0・・・カメラ装置、1 1・・・光学系、1 1 2・・・シフトレンズ、1 2・・・撮像素子、1 3・・・アナログフロントエンド回路（A F E）、1 4・・・姿勢センサ、1 4 1・・・加速度センサ、1 4 2・・・角速度センサ、1 6・・・システムコントローラ、1 6 1・・・画像信号処理部、1 6 2・・・マイクロコンピュータ、1 7・・・メモリ、1 8・・・表示装置、2 0・・・発音部。

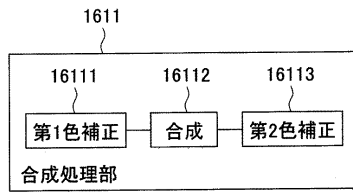
【 図 1 】



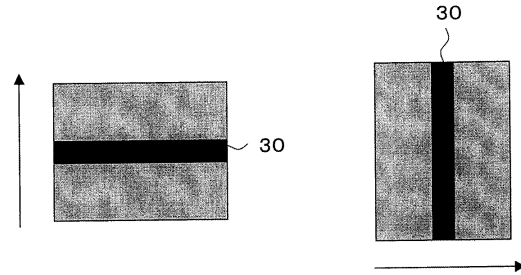
【 図 2 】



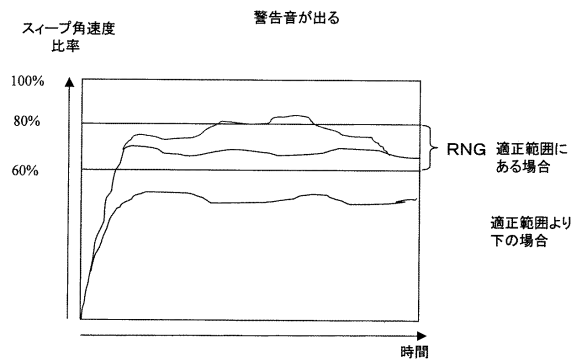
【図 3】



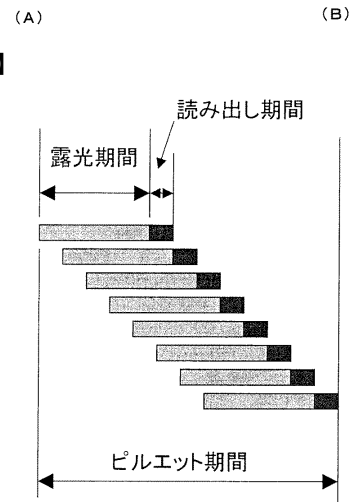
【図 5】



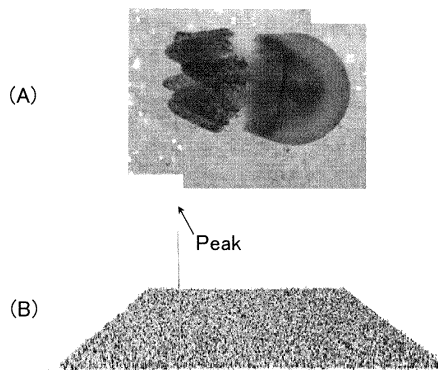
【図 4】



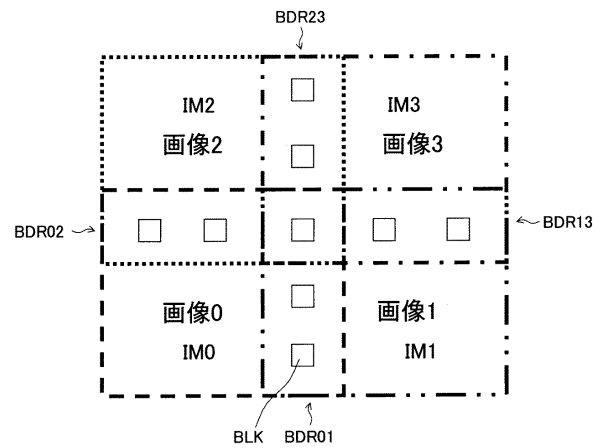
【図 6】



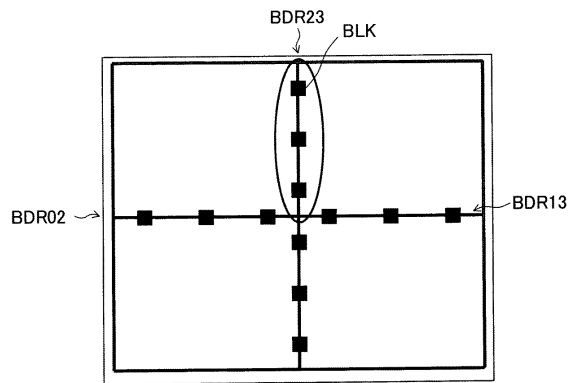
【図 7】



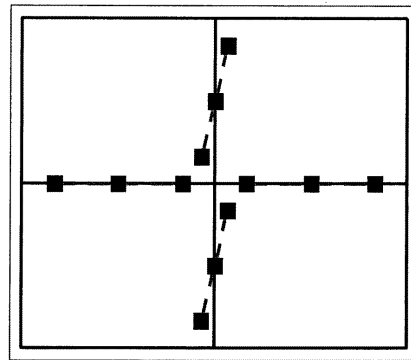
【図 8】



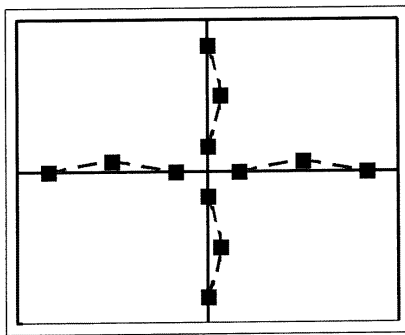
【図 9】



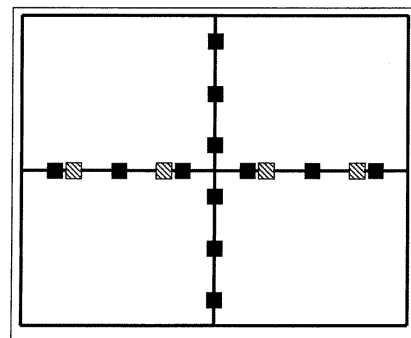
【図 11】



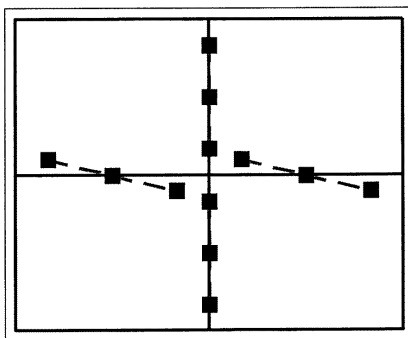
【図 10】



【図 12】

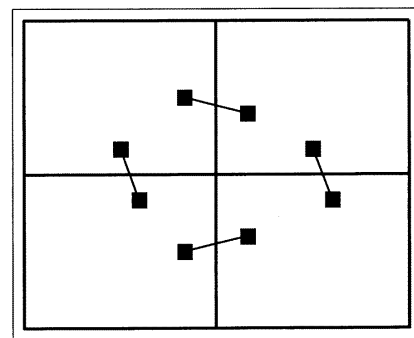


【図 13】

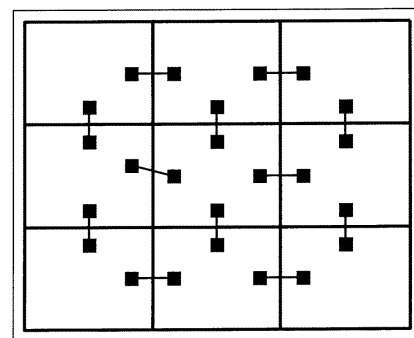


【図 14】

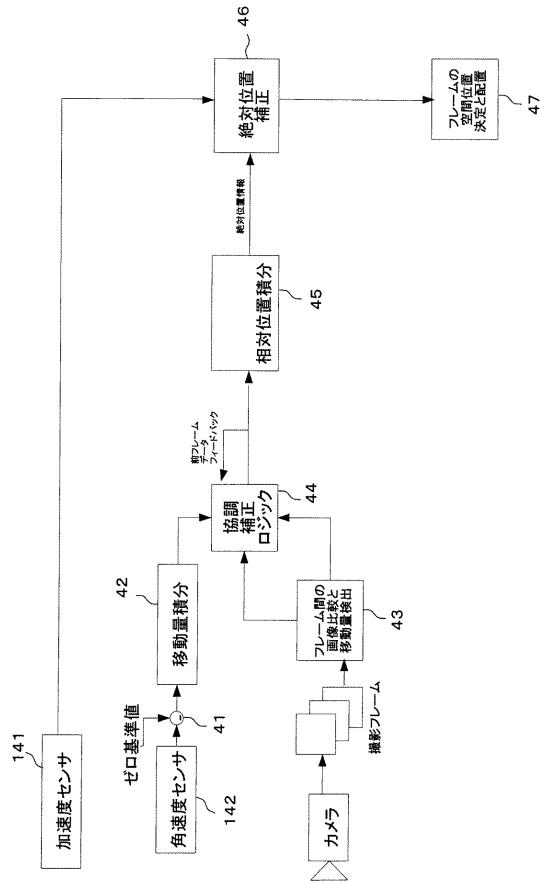
(A)



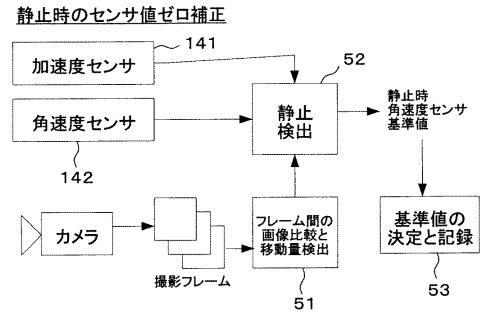
(B)



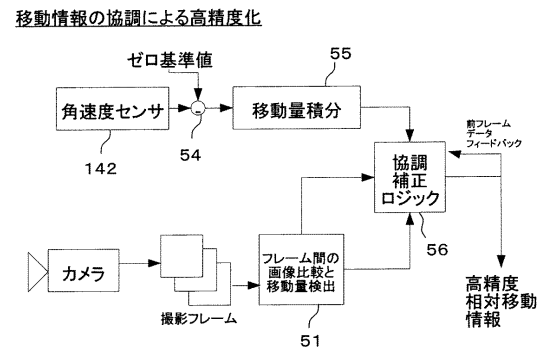
【図15】



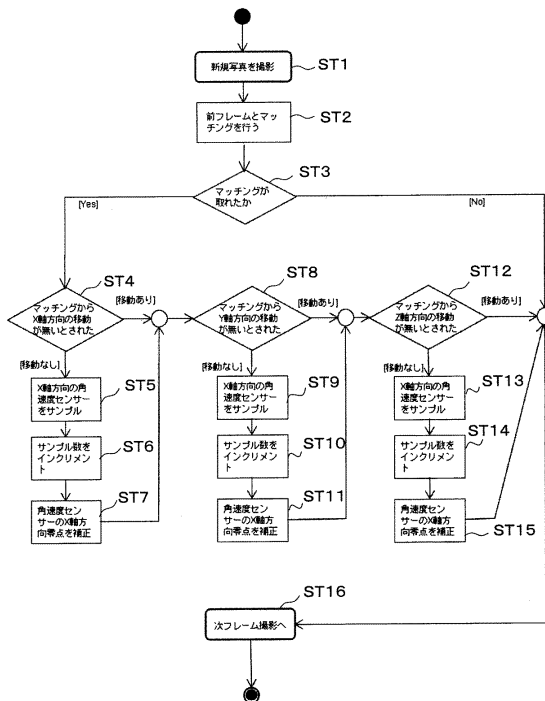
【図16】



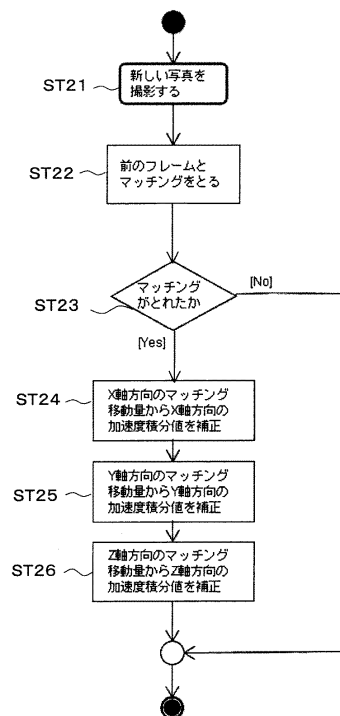
【図17】



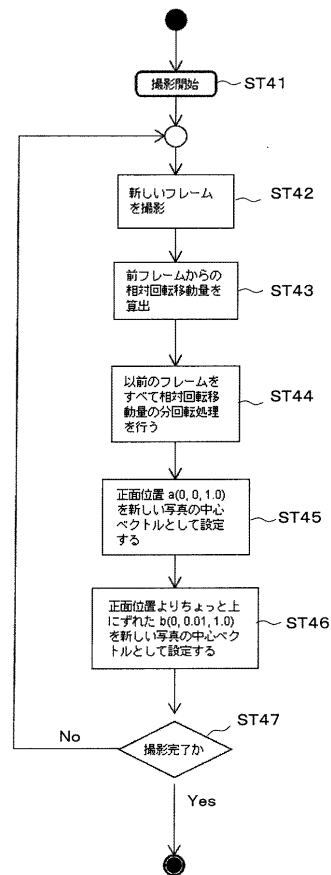
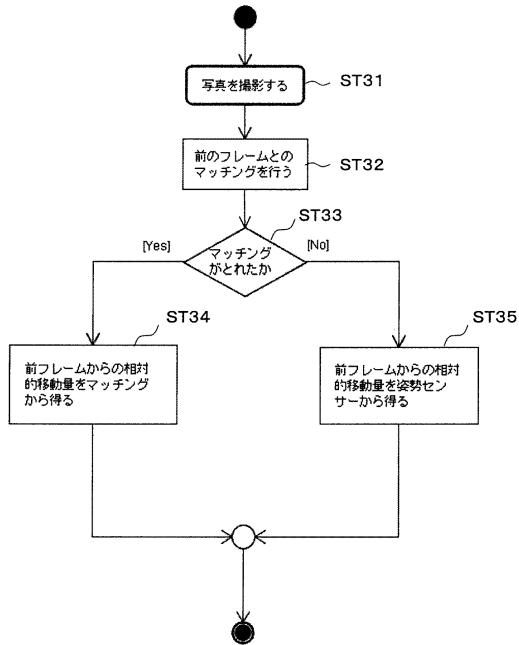
【図18】



【図19】



【 図 2 1 】



【 図 2 2 】

[illegible]

フロントページの続き

(56)参考文献 特開 2 0 0 8 - 1 1 7 2 3 5 (J P , A)
特開 2 0 0 6 - 2 1 7 4 7 8 (J P , A)
特開 2 0 0 5 - 3 2 1 9 0 2 (J P , A)
特開 2 0 0 5 - 3 2 8 4 9 7 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
H 0 4 N 5 / 2 3 2
H 0 4 N 5 / 2 2 5