

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4771186号
(P4771186)

(45) 発行日 平成23年9月14日(2011.9.14)

(24) 登録日 平成23年7月1日(2011.7.1)

(51) Int.Cl. F I
 G O 6 T 7/20 (2006.01) G O 6 T 7/20 1 0 0
 H O 4 N 5/232 (2006.01) H O 4 N 5/232 Z

請求項の数 13 (全 20 頁)

(21) 出願番号	特願2010-518003 (P2010-518003)	(73) 特許権者	304023318 国立大学法人静岡大学 静岡県静岡市駿河区大谷836
(86) (22) 出願日	平成21年6月22日 (2009.6.22)	(74) 代理人	100079049 弁理士 中島 淳
(86) 国際出願番号	PCT/JP2009/061329	(74) 代理人	100084995 弁理士 加藤 和詳
(87) 国際公開番号	W02009/154294	(74) 代理人	100099025 弁理士 福田 浩志
(87) 国際公開日	平成21年12月23日 (2009.12.23)	(72) 発明者	三浦 憲二郎 静岡県浜松市中区城北3丁目5-1 国立 大学法人静岡大学創造科学技術大学院内
審査請求日	平成22年12月8日 (2010.12.8)	(72) 発明者	▲高▼橋 賢治 静岡県浜松市中区城北3丁目5-1 国立 大学法人静岡大学工学部内
(31) 優先権主張番号	特願2008-162477 (P2008-162477)		最終頁に続く
(32) 優先日	平成20年6月20日 (2008.6.20)		
(33) 優先権主張国	日本国(JP)		
早期審査対象出願			

(54) 【発明の名称】 移動量抽出装置及びプログラム、画像補正装置及びプログラム並びに記録媒体

(57) 【特許請求の範囲】

【請求項1】

動画像を構成する複数のフレーム画像のうちの第1のフレーム画像に対して、平行移動量である第1方向への移動量 x 及び前記第1方向に直交する第2方向への移動量 y と、回転移動量 θ とを含んだアフィン変換パラメータを用いて画像変換処理を行って、第1の変換フレーム画像を生成する画像変換部と、

前記画像変換部により前記平行移動量及び前記回転移動量にそれぞれ所定の値が設定されて前記第1の変換フレーム画像が生成される毎に、前記画像変換部により生成された第1の変換フレーム画像と、前記動画像を構成する複数のフレーム画像のうちの前記第1のフレーム画像と異なる第2のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算し、前記第1の変換フレーム画像及び前記第2のフレーム画像が少なくとも重複するすべての同一座標についての前記自乗値を積算してエラー関数を導出するエラー関数導出部と、

BFGS法を用いて前記エラー関数導出部により導出されたエラー関数の値が最小値となる場合を探索し、前記エラー関数の値が最小値となる場合のアフィン変換パラメータを第2のフレーム画像に対する前記第1のフレーム画像の変化量として抽出する変化量抽出部と、

を備え、

前記変化量抽出部は、前記エラー関数の最小値の探索の際にBFGS法で使用する導関数として、

【数 1】

$$\begin{aligned}\frac{\partial E}{\partial \theta} &= -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right) \\ \frac{\partial E}{\partial b_1} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}} \\ \frac{\partial E}{\partial b_2} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}}\end{aligned}$$

10

但し、

$$\begin{aligned}\Delta I &= I^n(x_n) - I^{n+1}(x_{n+1}) \\ x_{n+1} &= x_n \cos \theta - y_n \sin \theta + b_1 \\ y_{n+1} &= x_n \sin \theta + y_n \cos \theta + b_2 \\ \frac{\partial x_{n+1}}{\partial \theta} &= -x_n \sin \theta - y_n \cos \theta \\ \frac{\partial y_{n+1}}{\partial \theta} &= x_n \cos \theta - y_n \sin \theta\end{aligned}$$

20

を用いる画像変化量抽出装置。

【請求項 2】

動画像を構成する複数のフレーム画像のうちの第 1 のフレーム画像に対して、平行移動量である第 1 方向への移動量 x 及び前記第 1 方向に直交する第 2 方向への移動量 y と、回転移動量 θ と、ズーム方向の倍率 z を含んだアフィン変換パラメータを用いて画像変換処理を行って、第 1 の変換フレーム画像を生成する画像変換部と、

30

前記画像変換部により前記平行移動量及び前記回転移動量にそれぞれ所定の値が設定されて前記第 1 の変換フレーム画像が生成される毎に、前記画像変換部により生成された第 1 の変換フレーム画像と、前記動画像を構成する複数のフレーム画像のうちの前記第 1 のフレーム画像と異なる第 2 のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算し、前記第 1 の変換フレーム画像及び前記第 2 のフレーム画像が少なくとも重複するすべての同一座標についての前記自乗値を積算してエラー関数を導出するエラー関数導出部と、

BFGS 法を用いて前記エラー関数導出部により導出されたエラー関数の値が最小値となる場合を探索し、前記エラー関数の値が最小値となる場合のアフィン変換パラメータを第 2 のフレーム画像に対する前記第 1 のフレーム画像の変化量として抽出する変化量抽出部と、

40

を備え、

前記変化量抽出部は、前記エラー関数の最小値の探索の際に BFGS 法で使用する導関数として、

【数 2】

$$\begin{aligned} \frac{\partial E}{\partial \theta} &= -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right) \\ \frac{\partial E}{\partial b_1} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}} \\ \frac{\partial E}{\partial b_2} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}} \\ \frac{\partial E}{\partial z} &= -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial z} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial z} \right) \end{aligned} \quad 10$$

但し、

$$\begin{aligned} \Delta I &= I^n(x_n) - I^{n+1}(x_{n+1}) \\ x_{n+1} &= zx_n \cos \theta - zy_n \sin \theta + b_1 \\ y_{n+1} &= zx_n \sin \theta + zy_n \cos \theta + b_2 \\ \frac{\partial x_{n+1}}{\partial \theta} &= -zx_n \sin \theta - zy_n \cos \theta \\ \frac{\partial y_{n+1}}{\partial \theta} &= zx_n \cos \theta - zy_n \sin \theta \\ \frac{\partial x_{n+1}}{\partial z} &= x_n \cos \theta - y_n \sin \theta \\ \frac{\partial y_{n+1}}{\partial z} &= x_n \sin \theta + y_n \cos \theta \end{aligned} \quad 20 \quad 30$$

を用いる画像変化量抽出装置。

【請求項 3】

前記エラー関数導出部は、前記第 1 の変換フレーム画像と、前記第 1 のフレーム画像に隣接する第 2 のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算する

請求項 1 又は 2 に記載の画像変化量抽出装置。

【請求項 4】

前記エラー関数導出部は、前記第 1 の変換フレーム画像と前記第 2 のフレーム画像との各々の同一座標について、各画素値の差分の自乗値をそれぞれ独立して並列演算する

請求項 1 又は 2 に記載の画像変化量抽出装置。

【請求項 5】

前記画像変換部は、動画像を構成する複数のフレーム画像のうちの最新の第 1 のフレーム画像に対して、前記画像変換処理を行って第 1 の変換フレーム画像を逐次生成し、前記エラー関数導出部は、前記画像変換部により逐次生成された第 1 の変換フレーム画像と、前記第 1 のフレーム画像の直前のフレームである第 2 のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算する

請求項 1 又は 2 に記載の画像変化量抽出装置。

【請求項 6】

請求項 1 又は 2 に記載の画像変化量抽出装置と、

前記第 1 のフレーム画像と前記画像変化量抽出装置により抽出された変化量とに基づいて、第 1 のフレーム画像と前記第 2 のフレーム画像とのずれが少なくなるように、前記第

1のフレーム画像に対して補正処理を行う補正部と、
を備えた画像補正装置。

【請求項7】

前記第2のフレーム画像に対して、前記補正部により補正処理された前記第1のフレーム画像を合成する画像合成部を更に備えた

請求項6に記載の画像補正装置。

【請求項8】

請求項6に記載の画像変化量抽出装置と、

前記第1のフレーム画像と前記画像変化量抽出装置により抽出された変化量とに基づいて、第1のフレーム画像と前記第2のフレーム画像とのずれが少なくなるように、前記第1のフレーム画像に対して補正処理を行う補正部と、

10

前記第2のフレーム画像に対して、前記補正部により補正処理された前記第1のフレーム画像を合成する画像合成部と、を備え、

前記画像変化量抽出装置は、次の第1のフレーム画像に対して、前記画像合成部により合成された画像を前記第2のフレーム画像として、前記次の第1のフレーム画像の変化量を抽出する

画像補正装置。

【請求項9】

請求項1又は2に記載の画像変化量抽出装置と、

前記第2のフレーム画像と前記画像変化量抽出装置により抽出された変化量とに基づいて、第1のフレーム画像と前記第2のフレーム画像とのずれが少なくなるように、前記第2のフレーム画像に対して補正処理を行う補正部と、

20

を備えた画像補正装置。

【請求項10】

前記第1のフレーム画像に対して、前記補正部により補正処理された前記第2のフレーム画像を合成する画像合成部を更に備えた

請求項9に記載の画像補正装置。

【請求項11】

コンピュータを、請求項6または請求項9に記載の画像補正装置の各部として機能させるための画像補正プログラム。

30

【請求項12】

コンピュータを、

動画像を構成する複数のフレーム画像のうちの第1のフレーム画像に対して、平行移動量である第1方向への移動量 x 及び前記第1方向に直交する第2方向への移動量 y と、回転移動量 θ とを含んだアフィン変換パラメータを用いて画像変換処理を行って、第1の変換フレーム画像を生成する画像変換手段、

前記画像変換手段より前記平行移動量及び前記回転移動量にそれぞれ所定の値が設定されて前記第1の変換フレーム画像が生成される毎に、前記画像変換手段により生成された第1の変換フレーム画像と、前記動画像を構成する複数のフレーム画像のうちの前記第1のフレーム画像と異なる第2のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算し、前記第1の変換フレーム画像及び前記第2のフレーム画像が少なくとも重複するすべての同一座標についての自乗値を積算してエラー関数を導出するエラー関数導出手段、及び、

40

BFGS法を用いて前記エラー関数導出手段により導出されたエラー関数の値が最小値となる場合を探索し、前記エラー関数の値が最小値となる場合のアフィン変換パラメータを第2のフレーム画像に対する前記第1のフレーム画像の変化量として抽出する変化量抽出手段、

として機能させるための画像変化量抽出プログラムであって、

前記変化量抽出手段は、前記エラー関数の最小値の探索の際にBFGS法で使用する導

50

関数として、
【数 3】

$$\begin{aligned}\frac{\partial E}{\partial \theta} &= -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right) \\ \frac{\partial E}{\partial b_1} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}} \\ \frac{\partial E}{\partial b_2} &= -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}}\end{aligned}$$

10

但し、

$$\begin{aligned}\Delta I &= I^n(x_n) - I^{n+1}(x_{n+1}) \\ x_{n+1} &= x_n \cos \theta - y_n \sin \theta + b_1 \\ y_{n+1} &= x_n \sin \theta + y_n \cos \theta + b_2 \\ \frac{\partial x_{n+1}}{\partial \theta} &= -x_n \sin \theta - y_n \cos \theta \\ \frac{\partial y_{n+1}}{\partial \theta} &= x_n \cos \theta - y_n \sin \theta\end{aligned}$$

20

を用いる画像変化量抽出プログラム。

【請求項 13】

コンピュータを、

動画を構成する複数のフレーム画像のうちの第 1 のフレーム画像に対して、平行移動量である第 1 方向への移動量 x 及び前記第 1 方向に直交する第 2 方向への移動量 y と、回転移動量 θ と、ズーム方向の倍率 z を含んだアフィン変換パラメータを用いて画像変換処理を行って、第 1 の変換フレーム画像を生成する画像変換手段、

30

前記画像変換手段より前記平行移動量及び前記回転移動量にそれぞれ所定の値が設定されて前記第 1 の変換フレーム画像が生成される毎に、前記画像変換手段により生成された第 1 の変換フレーム画像と、前記動画を構成する複数のフレーム画像のうちの前記第 1 のフレーム画像と異なる第 2 のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算し、前記第 1 の変換フレーム画像及び前記第 2 のフレーム画像が少なくとも重複するすべての同一座標についての自乗値を積算してエラー関数を導出するエラー関数導出手段、及び、

BFGS法を用いて前記エラー関数導出手段により導出されたエラー関数の値が最小値となる場合を探索し、前記エラー関数の値が最小値となる場合のアフィン変換パラメータを第 2 のフレーム画像に対する前記第 1 のフレーム画像の変化量として抽出する変化量抽出手段、

40

として機能させるための画像変化量抽出プログラムであって、

前記変化量抽出手段は、前記エラー関数の最小値の探索の際に BFGS法で使用する導関数として、

【数4】

$$\frac{\partial E}{\partial \theta} = -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right)$$

$$\frac{\partial E}{\partial b_1} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}}$$

$$\frac{\partial E}{\partial b_2} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}} \quad 10$$

$$\frac{\partial E}{\partial z} = -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial z} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial z} \right)$$

但し、

$$\Delta I = I^n(x_n) - I^{n+1}(x_{n+1})$$

$$x_{n+1} = zx_n \cos \theta - zy_n \sin \theta + b_1 \quad 20$$

$$y_{n+1} = zx_n \sin \theta + zy_n \cos \theta + b_2$$

$$\frac{\partial x_{n+1}}{\partial \theta} = -zx_n \sin \theta - zy_n \cos \theta$$

$$\frac{\partial y_{n+1}}{\partial \theta} = zx_n \cos \theta - zy_n \sin \theta$$

$$\frac{\partial x_{n+1}}{\partial z} = x_n \cos \theta - y_n \sin \theta$$

$$\frac{\partial y_{n+1}}{\partial z} = x_n \sin \theta + y_n \cos \theta \quad 30$$

を用いる画像変化量抽出プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、移動量抽出装置及びプログラム、画像補正装置及びプログラム並びに記録媒体に関する。

【背景技術】

【0002】

近年集積化技術の進歩により、ビデオカメラは、小型化・低価格化が進んで一般に普及するようになり、様々な場所で使用されている。特に、小型のビデオカメラは、近年、災害時に素早く情報収集を行うために、人間が立ち入れない場所で被災者の探索を行うロボットや、上空から災害状況を確認する無人ヘリコプターなど、遠隔操作のレスキューロボットなどにも搭載されている。

【0003】

しかし、ビデオカメラが搭載されたロボットは、自身が振動したり、荒れた路面上や地震で障害物が散在する状況下で走行したりする。このため、ロボットに搭載されたカメラから送られてくる映像にはゆれが生じてしまう。

【0004】

そのため、オペレータは、即座に状況判断をすることが困難になり、画面酔いして操作

10

20

30

40

50

に影響が出る可能性がある。したがって、このような映像のゆれによる影響を抑えるためには、リアルタイムで動画像処理が行われ、映像のゆれが軽減される必要がある。

【 0 0 0 5 】

現在、デジタルカメラのために開発・研究されているゆれを軽減する手法は、電子式、光学式、イメージセンサーシフト式、レンズユニットスイング式などの手ぶれ補正機能があげられる。しかし、これらの補正機能は、カメラに搭載されるものであり、そのカメラで撮影した映像だけが補正することができない。このため、必然的にカメラの大型化、高価格化を招いてしまう。

【 0 0 0 6 】

近年では、デジタルカメラの普及やパーソナルコンピュータ（PC）の発達により、一般の家庭用PCでも動画像の処理などが簡単に行われ、汎用性を高めるために、PCを利用した安定化処理が望まれる。しかしながら、動画像はデータ量が多く、それら进行处理するにはCPU（Central Processing Unit）では負荷が大きいため、リアルタイムでの処理は難しい。

【 0 0 0 7 】

そこで、高速なグラフィック処理を目的としたグラフィックスハードウェアであるGPU（Graphics Processing Unit）を用いることが考えられる。GPUは、一般的なPCにも搭載され、並列処理による高速演算が可能である。GPUの処理性能、特に浮動小数点演算性能は、CPUに比べて10倍以上の性能を示すものもある。

【 0 0 0 8 】

本願発明者らによる、GPUを用いたぶれ補正技術として「GPUを用いたビデオ映像の安定化」が開示されている（非特許文献1参照）。非特許文献1に記載された技術は、アフィン変換を用いてグローバルモーションを推定する際に、BFGS法（準ニュートン法）のアルゴリズムを使用し、推定したグローバルモーションに基づいてビデオ映像の振動を補正する。

【非特許文献1】藤澤、他2名、「GPUを用いたビデオ映像の安定化」、社団法人情報処理学会、情報処理学会論文誌Vol. 49、No. 2、p. 1 - 8

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 9 】

しかし、非特許文献1に記載された技術は、収束時間が長く、BFGS法の計算回数が多くなってしまいうため、グローバルモーション、すなわち、変化量の推定に時間がかかってしまう。このため、特許文献1の技術は、1秒当たり30枚のフレーム画像に対して4～5枚分しかぶれ補正処理をすることができないため、実質的にはリアルタイムで動画像のぶれ補正を行うことができなかった。

【 0 0 1 0 】

本発明は、上述した課題を解決するために提案されたものである。

【課題を解決するための手段】

【 0 0 1 1 】

本発明の第1の態様の画像変化量抽出装置は、動画像を構成する複数のフレーム画像のうち第1のフレーム画像に対して、平行移動量と回転移動量とを含んだアフィン変換パラメータを用いて画像変換処理を行って、第1の変換フレーム画像を生成する画像変換部と、前記画像変換部により前記平行移動量及び前記回転移動量にそれぞれ所定の値が設定されて前記第1の変換フレーム画像が生成される毎に、前記画像変換部により生成された第1の変換フレーム画像と、前記動画像を構成する複数のフレーム画像のうちの前記第1のフレーム画像と異なる第2のフレーム画像と、の同一座標の各画素値の差分の自乗値を演算し、前記第1の変換フレーム画像及び前記第2のフレーム画像が少なくとも重複するすべての同一座標についての自乗値を積算してエラー関数を導出するエラー関数導出部と、BFGS法を用いて前記エラー関数導出部により導出されたエラー関数の値が最小値となる場合を探索し、前記エラー関数の値が最小値となる場合のアフィン変換パラメータを

10

20

30

40

50

第2のフレーム画像に対する前記第1のフレーム画像の変化量として抽出する変化量抽出部と、を備えている。

【0012】

上記画像変化量抽出装置は、平行移動量及び回転移動量にそれぞれ所定の値が設定されて第1の変換フレーム画像が生成される毎に、第1の変換フレーム画像及び第2のフレーム画像が少なくとも重複するすべての同一座標についての自乗値を積算してエラー関数を導出し、BFGS法を用いてエラー関数の値が最小値となる場合を探索し、エラー関数の値が最小値となる場合のアフィン変換パラメータを第2のフレーム画像に対する前記第1のフレーム画像の変化量として抽出する。よって、探索時間を非常に短くして、第2のフレーム画像に対する前記第1のフレーム画像の変化量をリアルタイムで抽出できる。

10

【0013】

本発明の第2の態様の画像補正装置は、上記画像変化量抽出装置と、第1のフレーム画像と前記画像変化量抽出装置により抽出された変化量とに基づいて、第1のフレーム画像と前記第2のフレーム画像とのずれが少なくなるように、前記第1のフレーム画像に対して補正処理を行う補正部と、を備えている。

【0014】

また、本発明の第3の態様の画像補正装置は、上記画像変化量抽出装置と、第2のフレーム画像と前記画像変化量抽出装置により抽出された変化量とに基づいて、第1のフレーム画像と前記第2のフレーム画像とのずれが少なくなるように、前記第2のフレーム画像に対して補正処理を行う補正部と、を備えている。

20

【0015】

上記各画像補正装置は、リアルタイムで抽出された画像の変化量を用いて、その変化量に応じた画像の補正をリアルタイムで行うことができる。

【発明の効果】

【0016】

本発明の一態様である画像変化量抽出装置及びプログラムは、第1の変換フレーム画像及び第2のフレーム画像が少なくとも重複するすべての同一座標についての自乗値を積算してエラー関数を導出し、BFGS法を用いてエラー関数の値が最小値となる場合を探索して、エラー関数の値が最小値となる場合のアフィン変換パラメータを第2のフレーム画像に対する第1のフレーム画像の変化量として抽出する。これにより、エラー関数の値が最小値となる場合の探索時間を短くすることができ、動画像を構成する画像の変化量をリアルタイムで抽出することができる。

30

【0017】

本発明の一態様である画像補正装置及びプログラムは、動画像を構成する画像の変化量をリアルタイムで抽出することにより、その変化量に応じた画像の補正をリアルタイムで行うことができる。

【図面の簡単な説明】

【0018】

【図1】本発明の実施の形態に係る画像補正装置の構成を示すブロック図である。

【図2】グローバルモーションの推定を説明するための図である。

40

【図3】補正前と補正後（画像補正装置による補正済み）のフレーム数に対する移動量を示す図であり、（A）はX軸方向の移動量、（B）はY軸方向の移動量である。

【図4】第1から第3のフレーム画像の合成により生成された合成画像を示す図である。

【発明を実施するための最良の形態】

【0019】

以下、本発明の好ましい実施の形態について図面を参照しながら詳細に説明する。

【0020】

[第1の実施形態：画像補正装置の構成]

図1は、本発明の実施の形態に係る画像補正装置の構成を示すブロック図である。画像補正装置は、被写体を撮像して画像を生成するカメラ10と、カメラ10で生成された画

50

像のぶれをなくすように画像処理を行う画像処理装置 20 と、を備えている。

【0021】

画像処理装置 20 は、カメラ 10 との間で信号のやりとりを行う入出力ポート 21 と、演算処理を行う CPU (Central Processing Unit) 22 と、画像やその他のデータを記憶するハードディスクドライブ 23 と、CPU 22 の制御プログラムを記憶する ROM (Read Only Memory) 24 と、データのワークエリアである RAM (Random Access Memory) 25 と、画像処理のための所定の演算処理を行う GPU 26 (Graphics Processing Unit) と、を備えている。

【0022】

CPU 22 は、入出力ポート 21 を介して、カメラ 10 からの動画像を受信すると、動画像を GPU 26 に逐次転送し、GPU 26 に所定の演算処理を行わせて、動画像を構成する各フレーム画像から 1 フレーム毎にカメラ 10 の移動量を求める (グローバルモーションの推定)。なお、本実施形態では、振動の除去されたカメラ 10 の動きは緩やかで滑らかなものであると仮定する。そして、CPU 22 は、求めたカメラ 10 の移動量に基づいて、各フレーム画像に対して振動補正を行う。

10

【0023】

[グローバルモーションの推定]

映像の安定化を行うためにはグローバルモーションを知る必要がある。連続したフレーム間において、隣接したフレーム間での動きを求められれば、カメラ 10 がどのように動いたか知ることができる。

20

【0024】

隣接したフレーム画像 I^n と I^{n+1} の間の変換をアフィン変換であると仮定すると、ピクセル座標 $x = (x, y)$ の変化は式 (1) によって表すことができる。

【0025】

【数 1】

$$\mathbf{x}_{n+1} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{A}_n \mathbf{x}_n + \mathbf{b}_n \quad \dots (1)$$

また、式 (1) を変形すると式 (2) となる。

30

【0026】

【数 2】

$$\mathbf{x}_n = \mathbf{A}_{n-1} \mathbf{x}_{n-1} + \mathbf{b}_{n-1} = \dots = \left[\prod_{k=n}^1 \mathbf{A}_{k-1} \right] \mathbf{x}_1 + \sum_{k=1}^n \left[\prod_{m=n}^{k+1} \mathbf{A}_{m-1} \right] \mathbf{b}_k = \tilde{\mathbf{A}}_n \mathbf{x}_1 + \tilde{\mathbf{b}}_n \quad \dots (2)$$

【0027】

式 (2) は、任意のフレームからカメラ 10 がどのように動いたかを表す。アフィン変換パラメータ

40

【0028】

【数 3】

$$\left(\mathbf{A}_n^{n+1}, \mathbf{b}_n^{n+1} \right)$$

は、次の式 (3) のエラー関数の最小値 E_{\min} を求めることによって得られる。

【0029】

【数 4】

$$E(I^n, I^{n+1}, \mathbf{A}_n^{n+1}, \mathbf{b}_n^{n+1}) = \sum_{\mathbf{x} \in \mathcal{X}} (I^n(\mathbf{x}_n) - I^{n+1}(\mathbf{A}_n^{n+1} \mathbf{x}_n + \mathbf{b}_n^{n+1}))^2 \quad \dots (3)$$

【0030】

は画面平面上全ての座標値を表す。式(3)は2枚のフレーム画像の輝度値の差の2乗を合計した値である。ここで、非特許文献1に記載されたエラー関数

【0031】

【数 5】

$$E(I^n, I^{n+1}, \mathbf{A}_n^{n+1}, \mathbf{b}_n^{n+1}) = \sum_{\mathbf{x} \in \mathcal{X}} \sqrt{(I^n(\mathbf{x}_n) - I^{n+1}(\mathbf{A}_n^{n+1} \mathbf{x}_n + \mathbf{b}_n^{n+1})) + \beta}$$

と比較する。上記のエラー関数は、フレーム間の差分計算においてフレーム間の輝度差の絶対値を取得するために用いられる。

【0032】

仮に、上記式の絶対値を求めた場合(0)、合計計算をする前の式は厳密にフレーム間の差分画像の絶対値を得る。しかし、この式はルートを含んでおり、計算の上では非常に遅くなる。

【0033】

そこで、本実施形態の式(3)は、ルート計算及びを除いている。式(3)は、フレーム間の差分の2乗を合計したものであり、差分画像とは異なるものを表している。すなわち、式(3)が演算されても、人間が見ても何が写っているかわからない画像しか得られない。

【0034】

本来、グローバルモーションは人間も見える全体の動きである。よって、非特許文献1に記載されているように、エラー関数は素直に画像を重ね合わせた場合の画素値の差分の積算値である、と考えるのが最も自然である。

【0035】

これに対して、本実施形態の式(3)は、単なる2乗式であり、厳密には非特許文献1のエラー関数と同じ解が出るとは限らず、特殊であると考えられる。一方で、式(3)の解を用いて問題なく振動補正を行うことができる。すなわち、非特許文献1と本実施形態とでエラー関数の定義が異なるが、同じ結果を得ることができることがわかった。したがって、本実施形態の式(3)は、単なる2乗式にすることによって、ルートの計算がなくなる分計算が高速化することに加え、差分が大きくなり、最小値への収束が速くなること、グローバルモーション補正の失敗が少なくなる、という利点がある。そこで、図1に示す画像処理装置20のCPU22及びGPU26は、次のような演算を行う。

【0036】

図2は、グローバルモーションの推定を説明するための図である。カメラのぶれ量は、フレーム画像 I^n を基準にした場合のフレーム画像 I^{n+1} の画像移動量(回転角、x方向のそれぞれ移動量である b_1 、 b_2)となる。そこで、図1に示すCPU22は、フレーム画像 I^{n+1} の画像移動量の候補として、予め用意された複数のアフィン変換パラメータを記憶しており、フレーム画像 I^{n+1} と共に複数のアフィン変換パラメータをGPU26に送信する。なお、フレーム画像 I^{n+1} は、カメラ10で生成された動画像のうちの最新のフレーム画像であるのが好ましい。

【0037】

そして、CPU22は、GPU26に各アフィン変換パラメータが用いられた場合のエラー値 E を演算させて、エラー値 E が最小となる場合のアフィン変換パラメータをカメラ10の移動量として抽出する。なお、CPU22は、アフィン変換パラメータ(、 b_1 、

10

20

30

40

50

b 2) を単に GPU 26 に送信する代わりに、 から sin 及び cos を計算し、アフィン変換パラメータとして、b 1、b 2、sin 及び cos を GPU 26 に送信してもよい。

【0038】

一方、GPU 26 は、CPU 22 から送信されたアフィン変換パラメータを受信すると、フレーム画像 I^{n+1} に対して上述のアフィン変換パラメータを用いて変形処理を行う。

【0039】

具体的には、GPU 26 は、フレーム画像 I^n と変形されたフレーム画像 I^{n+1} との各々の同一座標についての画素値（輝度値）の差の 2 乗を演算する。なお、輝度値の差の 2 乗の演算は、すべての座標（例えば、少なくともフレーム画像 I^n 、 I^{n+1} の重複する領域のすべての座標）で行われる。なお、GPU 26 は、重複する領域での各同一座標において、輝度値の差の 2 乗値を並列かつ独立に演算する。これにより、GPU 26 は、各座標で独立に演算できるので、並列演算処理を行って高速処理を実行できる。そして、GPU 26 は、すべての座標における輝度値の差の 2 乗を並列積算し、積算値をエラー値として求める。ここでは、GPU 26 が、ある程度、輝度値の差の 2 乗を並列積算し、CPU 22 が、残りの輝度値の差の 2 乗を順次積算してこれらの積算値を合計するとよい。アフィン変換パラメータが変更される毎に上述したエラー値を演算する。

【0040】

ところで、フレーム画像 I^{n+1} を変形させたときの座標 (x', y') のピクセルとフレーム画像 I^n の座標 (x, y) のピクセルが対応している場合、その輝度値の差は 0 になり、エラー値は小さくなる。エラー値が小さい程フレーム間において対応するピクセルが多いということになり、そのときのパラメータ (A, b) がフレーム間の動きとなる。

【0041】

そこで、GPU 26 が、予め用意されたすべてのアフィン変換パラメータについて上述したエラー値を演算すると、次に、CPU 22 が、すべてのエラー値の中で最小のエラー値になったときのアフィン変換パラメータを選択し、選択したアフィン変換パラメータをフレーム間の動き、すなわちカメラの移動量として抽出する。

【0042】

なお、ピクセル座標値のアフィン変換

【0043】

【数 6】

$$A_n^{n+1} x_n + b_n^{n+1}$$

によって、輝度値が定義されていない領域（未定義域：フレーム画像 I^n 、 I^{n+1} の重複していない領域）を参照する場合、CPU 22 は、そのピクセルについてはエラー値の計算から除外するために、そのピクセルの輝度値の差を 0 とする。そして、CPU 22 は、全ピクセル に対し最終的に有効であったピクセル数 を用いて、次のようにエラー値 E を補正する。

【0044】

【数 7】

$$\hat{E} = (x/\chi_e)E$$

【0045】

ただし、 $\epsilon = \epsilon_0 / \epsilon_1$ が小さい場合（例えば $1 / 4$ ）に正しい結果が出ない可能性がある。例えば、実際のカメラ 10 の動きが小さくても、最小化手法の反復の初期において変形量が大きく、 ϵ の値が小さくなることがある。そこで、本実施形態では、CPU 22 は、 ϵ が $1 / 4$ 未満（ $< 1 / 4$ ）の場合、未定義域のピクセルの輝度値の差を 0 とみなしてエラー値を計算し、エラー値が意図的に大きくなるようにしている。なお、輝度値の差を 0 とみなすのは、 ϵ が 1 に比べて十分小さければよく、 ϵ が $1 / 4$ 未満の場合に限定されるものではない。

【0046】

また、エラー関数の最小値の探索には NUMERICAL RECIPES の BFGS 法（準ニュートン法）のアルゴリズムが使用される。BFGS（Broyden, Fletcher, Goldfarb, Shanno）法のアルゴリズムは関数と導関数を用いて最小方向の探索を行うので、計算回数が少なく、収束時間が短い。BFGS 法では導関数を必要とするので、導関数を求めるには式（3）を次の式（4）及び（5）のように変形する。

10

【0047】

【数8】

$$E = \sum_{x \in X} (I^n(x_n) - I^{n+1}(x_{n+1}))^2 = \sum_{x \in X} \Delta I^2 \quad \dots (4)$$

$$x_{n+1} = (a_1 x_n + a_2 y_n + b_1, a_3 x_n + a_4 y_n + b_2) \quad \dots (5)$$

20

【0048】

ここから導関数を求めると、式（6）となる。

【0049】

【数9】

$$\frac{\partial E}{\partial a_1} = \sum_{x \in X} 2\Delta I \frac{\partial \Delta I}{\partial a_1} \quad \dots (6)$$

30

【0050】

また、次の式（7）も成り立つ。

【0051】

【数10】

$$\frac{\partial \Delta I}{\partial a_1} = \frac{\partial \Delta I}{\partial I^{n-1}} \frac{\partial I^{n-1}}{\partial x_{n-1}} \frac{\partial x_{n-1}}{\partial a_1} = -1 \frac{\partial I^{n-1}}{\partial x_{n-1}} x_n \quad \dots (7)$$

40

【0052】

よって、すべての導関数は次の式（8）～（13）のようになる。

【0053】

【数 1 1】

$$\frac{\partial E}{\partial a_1} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial x_{n-1}} x_n \quad \dots (8)$$

$$\frac{\partial E}{\partial a_2} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial x_{n-1}} y_n \quad \dots (9)$$

$$\frac{\partial E}{\partial b_1} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial x_{n-1}} \quad \dots (10)$$

$$\frac{\partial E}{\partial a_3} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial y_{n-1}} x_n \quad \dots (11)$$

$$\frac{\partial E}{\partial a_4} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial y_{n-1}} y_n \quad \dots (12)$$

$$\frac{\partial E}{\partial b_2} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n-1}}{\partial y_{n-1}} \quad \dots (13)$$

10

【0054】

ここで、式(8)～(10)においては式(14)が成立し、式(11)～(13)においては(15)が成立する。

【0055】

【数 1 2】

$$\frac{\partial I^{n-1}}{\partial x_{n-1}} = \frac{I^{n+1}(x_{n+1} + \Delta x, y_{n+1}) - I^{n+1}(x_{n+1} - \Delta x, y_{n+1})}{2\Delta x} \quad \dots (14)$$

$$\frac{\partial I^{n-1}}{\partial y_{n-1}} = \frac{I^{n+1}(x_{n+1}, y_{n+1} + \Delta y) - I^{n+1}(x_{n+1}, y_{n+1} - \Delta y)}{2\Delta y} \quad \dots (15)$$

30

本実施形態では、高速化のため画像の動きを平行移動と回転移動のみと仮定して、求めるアフィン変換パラメータを(、 b_1 、 b_2)の3つとし、アフィン行列Tは式(16)とする。

【0056】

【数 1 3】

$$T = \begin{pmatrix} \cos\theta & \sin\theta & b_1 \\ \sin\theta & \cos\theta & b_2 \\ 0 & 0 & 1 \end{pmatrix} \quad \dots (16)$$

40

また、導関数は、次の式(17)～(19)とする。

【0057】

【数 1 4】

$$\frac{\partial E}{\partial \theta} = -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right) \dots (17)$$

$$\frac{\partial E}{\partial b_1} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}} \dots (18)$$

$$\frac{\partial E}{\partial b_2} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}} \dots (19)$$

10

【0058】

ここで、式(*)と定義したとき式(20)~(23)とする。

【0059】

【数 1 5】

$$\Delta I = I^n(\mathbf{x}_n) - I^{n+1}(\mathbf{x}_{n+1}) \dots (*)$$

$$x_{n+1} = x_n \cos \theta - y_n \sin \theta + b_1 \dots (20)$$

20

$$y_{n+1} = x_n \sin \theta + y_n \cos \theta + b_2 \dots (21)$$

$$\frac{\partial x_{n+1}}{\partial \theta} = -x_n \sin \theta - y_n \cos \theta \dots (22)$$

$$\frac{\partial y_{n+1}}{\partial \theta} = x_n \cos \theta - y_n \sin \theta \dots (23)$$

【0060】

すなわち、図1に示す画像処理装置20のCPU22は、式(16)のアフィン変換行列を用いて式(3)のエラー関数を定義し、このエラー関数の最小値を探索するために準ニュートン法の手法の一つであるBFGS法を用いる。ここで、BFGS法では導関数が必要となる。そこで、CPU22は、式(17)~(19)(式(20)~(23)を含む。)の導関数を用いて式(3)のエラー関数の最小値を探索し、最小値となったときのパラメータ(、 b_1 、 b_2)を求め、これを画像移動量、すなわちカメラ10のぶれ量として抽出する。

30

【0061】

なお、複数回のエラー関数の導出においては、1つのエラー関数が導出されると、新たなアフィン変換パラメータ(、 b_1 、 b_2 の少なくとも1つが所定量変化したもの)を用いて再びエラー関数が導出される。なお、これらのパラメータの変更の仕方は特に限定されるものではない。また、BFGS法については、Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: Numerical Recipes in C++: The Art of Scientific Computing, Cambridge University Press (2002)に記載された手法を用いることができる。

40

〔振動補正〕

画面の動きを滑らかにするために、推定したグローバルモーションに基づく修正のための変換行列を求める必要がある。補正前のフレームから補正後のフレームまでの変換行列Sは、補正の対象となるフレームの前後kフレームまでのアフィン変換を用いて次の式(24)によって表される。

【0062】

【数 16】

$$S_n = \sum_{m=n-k}^{n+k} T_n^m \star G(k) \quad \dots (24)$$

ここで、

【0063】

【数 17】

$$T_n^m$$

10

はフレーム n から m までのアフィン変換行列である。また、

【0064】

【数 18】

$$G(k) = \frac{1}{\sqrt{2\pi\sigma}} e^{-k^2/2\sigma^2}$$

20

はガウス・カーネルである。式 (24) の星印は畳み込み演算子を表す。また $k =$ である。

【0065】

そこで、図 1 に示す画像処理装置 20 の CPU 22 は、得られた変換行列を用いて次の式 (25) の演算を行うことで、フレーム画像間のずれが少なくなるように、対象となるフレーム画像に対して振動補正を行うことができる。

【0066】

【数 19】

$$\left. \begin{aligned} \bar{x}_n &= S_n x_n \\ &= \hat{A}_n x_1 + \hat{b}_n \\ &= (\hat{A}_n (\tilde{A})_n^{-1}) x_n + (\hat{A}_n (\tilde{A})_n^{-1}) \tilde{b}_n + \hat{b}_n \\ &= \bar{A}_n x_n + \bar{b}_n \end{aligned} \right\} \dots (25)$$

30

【0067】

ここで、隣り合うフレーム画像間の振動補正を行う場合は上述した n 、 m は連続した自然数となるが、基準となるフレーム画像に対して所定のフレーム画像の振動補正を行う場合は、 n 、 m は連続した自然数でなくてもよい。

40

【0068】

なお、本願発明者らが 1 フレームあたりの BFGS 法の適用回数を計算したところ次の結果が得られた。非特許文献 1 に記載されたエラー関数を用いると、GPU が演算した場合は平均で 42.87 回、CPU が演算した場合は平均で 11.43 回となった。これに対して、本実施形態の式 (3) のエラー関数を用いると、GPU が演算した場合は平均で 7.707 回、CPU が演算した場合は平均で 6.481 回となった。すなわち、式 (3) のエラー関数を用いた方が、演算回数が少なくなり、短時間で演算することが可能になる。

【0069】

図 3 は、補正前と補正後 (画像補正装置による補正済み) のフレーム数に対する移動量

50

を示す図であり、(A)はX軸方向の移動量、(B)はY軸方向の移動量である。同図に示すように、補正により移動量は非常に滑らかになった。

【0070】

また、画像処理装置20のCPU22は、回転移動量及び平行移動量の少なく1つが補正されたフレーム画像を順次合成して、複数のフレームからなる合成画像を生成してもよい。

【0071】

図4は、第1から第3のフレーム画像の合成により生成された合成画像を示す図である。ここでは、CPU22は、補正された最新のフレーム画像を、中心位置に対して水平になるように順次に重ねる。これにより、中心付近は新しいフレーム画像で構成され、縁付近は古いフレーム画像で構成された、フレーム画像よりも大きい合成画像が生成される。

10

【0072】

この場合、GPU26は、各座標において画像が存在するかの判定フラグを立てて、画像がある座標においてのみエラー関数Eを演算すればよい。この結果、フレーム画像の移動量の推定誤差が小さくなり、仮に、最新のフレーム画像とその1つ前のフレーム画像との重なる部分がほとんどなくても、グローバルモーション推定が可能になる。なお、GPU26は、累積誤差を防止するため、最新のフレーム画像よりも所定フレーム以前のフレーム画像については合成せず、順次破棄するとよい。

20

【0073】

さらに、GPU26は、過去のフレーム画像 I^n 、 I^{n-1} 、 I^{n-2} ・・・の合成である合成画像をフレーム画像 I^n とし、次の最新のフレーム画像 I^{n+1} を用いて、エラー関数Eを演算してもよい。これにより、カメラ10のぶれ量が大きい場合でも、合成されたフレーム画像 I^n と次の最新のフレーム画像 I^{n+1} との重複範囲が大きくなるので、カメラのぶれ量が確実に検出される。

20

【0074】

以上のように、本発明の実施形態に係る画像補正装置は、BFGS法を適用して式(3)のエラー関数の最小値を探索することにより、従来に比べて、非常に短時間でエラー関数が最小値となる時のアフィン変換パラメータを求め、このアフィン変換パラメータを用いて、リアルタイムで動画像のぶれを補正することができる。

30

【0075】

BFGS法を用いた最小値探索では複数回の演算を繰り返して最小値を探索するので、個々の演算式の演算速度のわずかな違いでも、最終的な演算速度に大きく響いてくる。特に、本実施形態に係る画像補正装置は、画像の各画素についての演算を行うため、この違いは顕著である。非特許文献1では、個々の演算式に平方根が含まれているため、どうしても演算速度が遅くなってしまふ。これに対して、本実施形態に係る画像補正装置は、エラー関数を工夫することにより、平方根の演算を用いることなく高速にエラー関数の最小値を探索することが可能となった。また、エラー関数を用いることにより、BFGS法を用いた最小値探索の繰り返し演算回数そのものを少なくできることも見出した。

40

【0076】

さらに、上記画像補正装置は、補正されたフレーム画像を順次合成することで、フレーム画像よりもサイズの大きい合成画像を生成することができる。そして、上記画像補正装置は、サイズの大きい合成画像に対する最新のフレーム画像の移動量を抽出することで、カメラ10のぶれ量が大きい場合でも、そのぶれ量を確実に抽出して、ぶれを補正することができる。

【0077】

なお、上記画像補正装置は、カメラ10がぶれた場合だけでなく、被写体がぶれた場合であっても、上述した式(3)を用いてリアルタイムで動画像の被写体ぶれを補正することができる。

【0078】

50

[第2の実施形態：他のアフィン変換パラメータを用いた場合]

つぎに、本発明の第2の実施形態について説明する。なお、第1の実施形態と同一の部位には同一の符号を付し、重複する説明は省略する。

【0079】

第1の実施形態では3変数であるアフィン変換パラメータ(、 b_1 、 b_2)を用いたが、第2の実施形態では4変数であるアフィン変換パラメータ(、 b_1 、 b_2 、 z)を用いる。なお、 z はズーム方向のパラメータであり、画像の倍率を示している。ここで、エラー関数は、次の式(26)のように表される。

【0080】

【数20】

10

$$E(n, n+1) = \sum_{x \in X} (I^n(x_n) - I^{n+1}(A_n^{n+1}x_n + b_n^{n+1}))^2 \quad \dots (26)$$

【0081】

式(26)において X は画面平面上全ての座標値の集合である。 $I(x)$ はピクセル x の輝度値である。そして、4変数のアフィン変換パラメータを用いると、アフィン変換は、次の式(27)のように表される。

20

【0082】

【数21】

$$x_{n+1} = A_n^{n+1}x_n + b_n^{n+1} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \dots (27)$$

【0083】

このとき、導関数は、式(28)～(31)のようになる。

【0084】

30

【数22】

$$\frac{\partial E}{\partial \theta} = -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial \theta} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial \theta} \right) \quad \dots (28)$$

$$\frac{\partial E}{\partial b_1} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial x_{n+1}} \quad \dots (29)$$

$$\frac{\partial E}{\partial b_2} = -2 \sum_{x \in X} \Delta I \frac{\partial I^{n+1}}{\partial y_{n+1}} \quad \dots (30)$$

$$\frac{\partial E}{\partial z} = -2 \sum_{x \in X} \Delta I \left(\frac{\partial I^{n+1}}{\partial x_{n+1}} \frac{\partial x_{n+1}}{\partial z} + \frac{\partial I^{n+1}}{\partial y_{n+1}} \frac{\partial y_{n+1}}{\partial z} \right) \quad \dots (31)$$

40

【0085】

但し、導関数は、式(32)～(38)を満たす。

【0086】

【数 2 3】

$$\Delta I = I^n(\mathbf{x}_n) - I^{n+1}(\mathbf{x}_{n+1}) \quad \dots (32)$$

$$x_{n+1} = zx_n \cos \theta - zy_n \sin \theta + b_1 \quad \dots (33)$$

$$y_{n+1} = zx_n \sin \theta + zy_n \cos \theta + b_2 \quad \dots (34)$$

$$\frac{\partial x_{n+1}}{\partial \theta} = -zx_n \sin \theta - zy_n \cos \theta \quad \dots (35)$$

$$\frac{\partial y_{n+1}}{\partial \theta} = zx_n \cos \theta - zy_n \sin \theta \quad \dots (36)$$

$$\frac{\partial x_{n+1}}{\partial z} = x_n \cos \theta - y_n \sin \theta \quad \dots (37)$$

$$\frac{\partial y_{n+1}}{\partial z} = x_n \sin \theta + y_n \cos \theta \quad \dots (38)$$

10

【0087】

そこで、第2の実施形態では、図1に示す画像処理装置20のCPU22は、上述した4変数のアフィン変換パラメータを用いたエラー関数に対して、式(28)~(31)(式(32)~(38))を含む。)を用いてBFGS法を適用する。これにより、CPU22は、エラー値の最小値を短時間で探索し、このときのアフィン変換パラメータをフレーム間の動き、すなわちカメラの移動量として抽出する。そして、CPU22は、このアフィン変換パラメータを用いて第1の実施形態と同様にして画像を補正することができる。

20

【0088】

以上のように、第2の実施形態に係る画像補正装置は、ズーム方向のパラメータを含んだアフィン変換パラメータを用いて移動量を抽出できるので、画像に現れる被写体の大きさが変化するほどカメラ10が振動しても、リアルタイムでその振動を抑制するように動画像を補正することができる。

【0089】

また、本発明は、上述した実施の形態に限定されるものではなく、特許請求の範囲に記載された範囲内で設計上の変更をされたものにも適用可能であるのは勿論である。例えば、上述した実施形態では、フレーム画像 I^n に隣接したフレーム画像 I^{n+1} の変換をアフィン変換パラメータで表したが、変換されるフレーム画像はフレーム画像 I^n に隣接していなくてもよい。例えば、基準フレーム画像に対して数フレーム離れた所定のフレーム画像をアフィン変換パラメータで表すことも可能である。

30

【0090】

また、上述した実施形態では、画像処理装置20は、カメラ10で生成された動画像をリアルタイムで補正したが、ハードディスクドライブ23に予め記憶されている動画像についても同様に補正することができる。

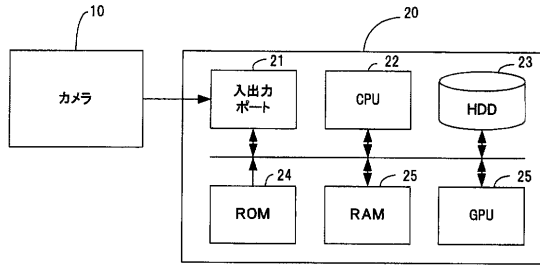
【符号の説明】

【0091】

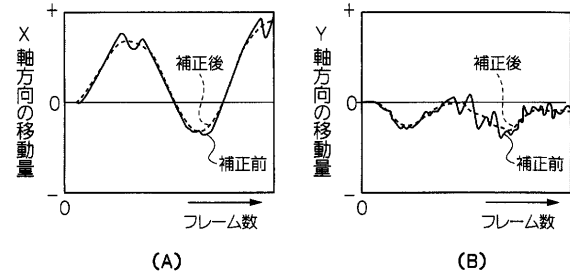
- 10 カメラ
- 20 画像処理装置
- 22 CPU
- 26 GPU

40

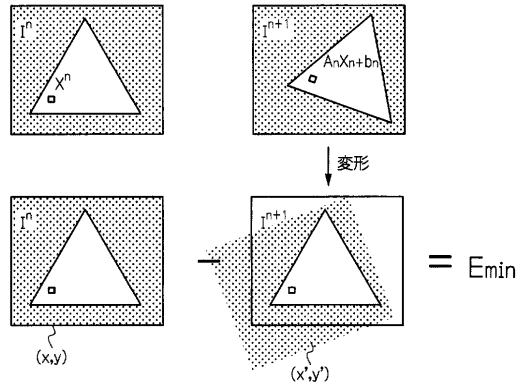
【図1】



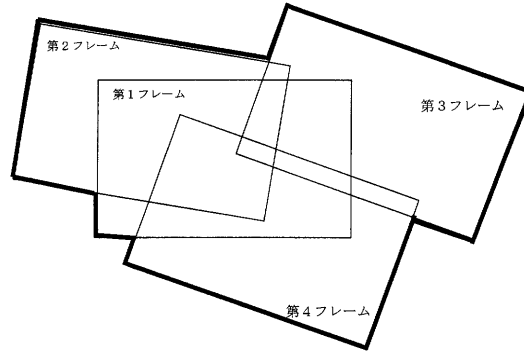
【図3】



【図2】



【図4】



フロントページの続き

審査官 松永 稔

- (56)参考文献 特開2005-309558(JP,A)
特開2007-041752(JP,A)
特開2005-331440(JP,A)
特開2007-035020(JP,A)
特表2006-520042(JP,A)
藤澤誠, 外2名, GPUを用いたビデオ映像の安定化, 情報処理学会論文誌, 日本, 社団法人情報処理学会, 2008年 2月15日, 第49巻, 第2号, p.1022-1030

(58)調査した分野(Int.Cl., DB名)

G06T 7/20

H04N 5/232