

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2013-50947
(P2013-50947A)

(43) 公開日 平成25年3月14日(2013.3.14)

(51) Int.Cl. F I テーマコード(参考)
G06T 1/00 (2006.01) G06T 1/00 315 3C707
B25J 19/04 (2006.01) B25J 19/04 5B057

審査請求 未請求 請求項の数 20 O L (全 30 頁)

(21) 出願番号 特願2012-162378 (P2012-162378)
 (22) 出願日 平成24年7月23日(2012.7.23)
 (31) 優先権主張番号 13/220,851
 (32) 優先日 平成23年8月30日(2011.8.30)
 (33) 優先権主張国 米国(US)

(71) 出願人 000002369
 セイコーエプソン株式会社
 東京都新宿区西新宿2丁目4番1号
 (74) 代理人 100095728
 弁理士 上柳 雅誉
 (74) 代理人 100107261
 弁理士 須澤 修
 (74) 代理人 100127661
 弁理士 宮坂 一彦
 (72) 発明者 アラシュ アバドゥポア
 カナダ国 オンタリオ州 トロント ハリ
 ソン ガーデン ブルバード 16-L p
 h07

最終頁に続く

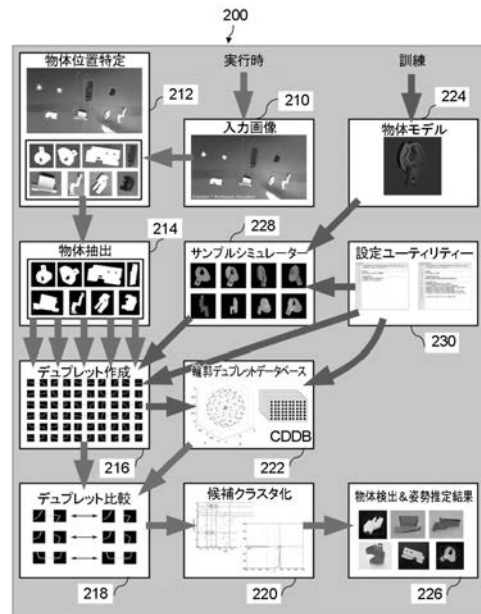
(54) 【発明の名称】 物体姿勢推定方法、物体姿勢推定装置、物体推定姿勢精緻化方法、およびコンピューター可読媒体

(57) 【要約】 (修正有)

【課題】 物体を、ロボットの搭載カメラから見える通りのその姿勢にかかわらず認識することができるようにする。

【解決手段】 物体を包含する画像を入力し、入力画像の2値マスクを作成、入力画像の2値マスクからシングレットのセットを抽出する。各シングレットは、入力画像内の物体の内側および外側輪郭内にある点を表す、抽出することと、シングレットのセットを連結し、デュプレックス行列として表されるメッシュにし、候補姿勢のセットを作るべく2つのデュプレックス行列を比較するこし、物体姿勢推定値によって、入力画像から物体の姿勢が推定され、次に、物体姿勢推定が記憶される。物体の推定姿勢は、カメラのパラメーターの入力、物体のモデルを物体の仮想画像に投影、初期姿勢パラメーターを新しい姿勢パラメーターに更新、エネルギー関数の最小化によって精緻化される。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

入力画像から物体の姿勢を推定して記憶する物体姿勢推定方法であって、
物体を包含する画像を入力することと、
前記入力画像の 2 値マスクを作成することと、
前記入力画像の前記 2 値マスクからシングレットのセットを抽出することであって、各
シングレットは、前記入力画像内の前記物体の内側および外側輪郭内にある点を表す、抽
出することと、
シングレットの前記セットを連結して、デュプレックス行列として表されるメッシュに
することと、
候補姿勢のセットを作るべく 2 つのデュプレックス行列を比較することと、
物体姿勢推定値を作り、前記物体姿勢推定値を記憶することと、
を含む、物体姿勢推定方法。

10

【請求項 2】

前記物体の複数の画像を入力することをさらに含み、各画像は、前記複数の画像の互い
における視像とは異なる、前記物体の視像を包含する、請求項 1 に記載の物体姿勢推定方
法。

【請求項 3】

前記物体の前記複数の画像の各々における前記視像を、前記物体のための C A D モデル
を用いて生成することをさらに含む、請求項 2 に記載の物体姿勢推定方法。

20

【請求項 4】

前記物体の前記複数の画像の各々における前記視像を、カメラを備えたロボットを用い
て生成することをさらに含む、請求項 2 に記載の物体姿勢推定方法。

【請求項 5】

前記入力画像内の前記物体を検出し、前記物体のバウンディングボックスを計算するこ
とをさらに含む、請求項 1 に記載の物体姿勢推定方法。

【請求項 6】

前記物体の内側および外側輪郭を抽出することをさらに含む、請求項 1 に記載の物体姿
勢推定方法。

【請求項 7】

前記姿勢推定値をエネルギー関数の最適化としてモデル化することによって前記物体姿
勢推定値を精緻化することをさらに含む、請求項 1 に記載の物体姿勢推定方法。

30

【請求項 8】

前記エネルギー関数におけるエネルギー値を最小化するべく姿勢推定値を反復的に計算
することをさらに含む、請求項 7 に記載の物体姿勢推定方法。

【請求項 9】

前記姿勢推定値を反復的に計算するために速度スクリューを計算することをさらに含む
、請求項 8 に記載の物体姿勢推定方法。

【請求項 10】

レンダリングアプリケーションプログラミングインタフェース (a p p l i c a t i o
n p r o g r a m m i n g i n t e r f a c e、A P I) を用いて前記物体のモデル
輪郭曲線を仮想画像に投影することをさらに含む、請求項 7 に記載の物体姿勢推定方法。

40

【請求項 11】

入力画像から物体の姿勢を推定して記憶する物体姿勢推定装置であって、
プロセッサであって、
物体を包含する入力画像を受信すること、
前記入力画像の 2 値マスクを作成すること、
前記入力画像の前記 2 値マスクからシングレットのセットを抽出することであって、
各シングレットは、前記入力画像内の前記物体の内側および外側輪郭内にある点を表す、
抽出すること、

50

シングレットの前記セットを連結して、デュプレックス行列として表されるメッシュにすること、

候補姿勢のセットを作るべく2つのデュプレックス行列を比較すること、ならびに物体姿勢推定値を作ること、

を行うプロセッサと、

前記物体姿勢推定値を記憶するメモリーと、

を含む、物体姿勢推定装置。

【請求項12】

前記プロセッサは前記物体の複数の画像を受信し、各画像は、前記複数の画像の互いにおける視像とは異なる、前記物体の視像を包含する、請求項11に記載の物体姿勢推定装置。

10

【請求項13】

前記物体のためのCADモデルを記憶するストレージをさらに含み、前記プロセッサは、前記物体のための前記CADモデルを用いて前記物体の前記複数の画像の各々における前記視像を生成する、請求項12に記載の物体姿勢推定装置。

【請求項14】

前記物体の前記複数の画像の各々における前記視像を生成するカメラを備えたロボットをさらに含み、請求項12に記載の物体姿勢推定装置。

【請求項15】

物体の推定姿勢を精緻化する物体姿勢推定精緻化方法であって、

20

推定姿勢をとる物体の画像、前記物体のモデル、および前記推定姿勢をとる前記物体の前記画像を撮るために用いられるカメラのパラメータを入力することと、

2値マスク画像および画像深さ情報を得るべく、前記カメラの前記パラメータおよび初期姿勢パラメータを用いて前記物体の前記モデルを前記物体の仮想画像に投影することと、

前記2値マスク画像および画像深さ情報を用いて前記初期姿勢パラメータを新しい姿勢パラメータに更新し、エネルギー関数を最小化するべくまたは反復の最大数に達するまで前記新しい姿勢パラメータを反復的に更新することと、

を含む、物体姿勢推定精緻化方法。

【請求項16】

30

コスト関数を用いて前記物体の内部および外部の平均値を計算することと、

前記2値マスク画像から前記物体の輪郭Cを計算し、前記画像深さ情報を用いて3D輪郭点を計算することと、

をさらに含む、請求項15に記載の物体推定姿勢精緻化方法。

【請求項17】

前記輪郭点についての、 x および y 方向における勾配 $T(x, y)$ を計算することと

、画像ヤコビ行列を計算することと、

ガウス・ニュートン法を用いて勾配流および速度スクリューを計算することと、

をさらに含む、請求項16に記載の物体推定姿勢精緻化方法。

40

【請求項18】

前記速度スクリューおよび所与のステップワイズを用いて相対変位を計算すること

をさらに含む、請求項17に記載の物体推定姿勢精緻化方法。

【請求項19】

前記推定姿勢をとる前記物体の前記画像は、

前記物体を包含する画像を入力することと、

前記入力画像の2値マスクを作成することと、

前記入力画像の前記2値マスクからシングレットのセットを抽出することであって、各シングレットは、前記入力画像内の前記物体の内側および外側輪郭内にある点を表す、抽出することと、

50

シングレットの前記セットを連結して、デュプレックス行列として表されるメッシュにすることと、

候補姿勢のセットを作るべく2つのデュプレックス行列を比較することと、

物体姿勢推定値を作ることと、

によって形成される、請求項15に記載の物体推定姿勢精緻化方法。

【請求項20】

1つ以上の有形の非一時的コンピューター可読媒体であって、プロセッサによって実行されると請求項1に記載の物体姿勢推定方法に従い入力画像から物体の姿勢を推定するコンピューター可読命令をその上に有する、コンピューター可読媒体。

【発明の詳細な説明】

10

【技術分野】

【0001】

本発明は、物体の姿勢を推定する物体の物体姿勢推定方法、物体姿勢推定装置、物体推定姿勢精緻化方法、およびコンピューター可読媒体に関する。

【背景技術】

【0002】

今日の多くの工業環境では、部品の組み立ておよび製造のためにロボットが用いられている。これらのロボットは、それらに視覚を与える1つ以上のカメラ、例えばCCDおよびCMOS、を備えている。多くの場合、物体（すなわち部品）は容器に入っている。ロボットは、製品を組み立てるべく物体を拾い上げることができるように、容器内の物体/部品を認識しなければならない。しかし、物体は様々な姿勢（位置、配向、回転）をとり得る。従って、ロボットは部品をその姿勢にかかわらず認識するように訓練されなければならない。

20

本発明は、ロボットが訓練されて、物体を、ロボットの搭載カメラから見える通りのその姿勢にかかわらず認識することができるよう、2次元画像（例えばカメラ画像）から3次元（three-dimensional、3D）物体の姿勢を推定することに関する。

【発明の概要】

【発明が解決しようとする課題】

【0003】

30

本技術分野において周知であるように、ロボットは、カメラの画像から物体を特定しようとするソフトウェアを含む。従って、このようなソフトウェアは、ロボットのカメラによって取り込まれた画像と比較するために用いる物体画像のロバスト且つ正確なデータベースを有することが重要である。

【課題を解決するための手段】

【0004】

本発明の1つの態様は、入力画像から物体の姿勢を推定し、物体姿勢推定を記憶する方法および装置であって、該方法および装置は、物体を包含する画像を入力することと、入力画像の2値マスクを作成することと、入力画像の2値マスクからシングレットのセットを抽出することと、各シングレットは、入力画像内の物体の内側および外側輪郭内にある点を表す、抽出することと、シングレットのセットを連結して、デュプレックス行列として表されるメッシュにすることと、候補姿勢のセットを作るべく2つのデュプレックス行列を比較することと、物体姿勢推定値を作り、該物体姿勢推定値を記憶することと、を含む、方法および装置である。

40

【0005】

本発明の一実施形態では、物体の複数の画像が入力され、各画像は、該複数の画像の互いにおける視像とは異なる、物体の視像を包含する。

1つの態様は、物体の複数の画像の各々における視像を、物体のためのCADモデルを用いて生成することを含む。

別の態様は、物体の複数の画像の各々における視像を、カメラを備えたロボットを用い

50

て生成することを含む。

【0006】

本発明のさらなる態様は、入力画像内の物体を検出し、物体のバウンディングボックスを計算することを含む。

別の態様は、物体の内側および外側輪郭を抽出することを含む。

【0007】

本発明の他の実施形態では、物体姿勢推定値は、該姿勢推定値をエネルギー関数の最適化としてモデル化することによって精緻化される。

1つの態様では、本発明は、エネルギー関数におけるエネルギー値を最小化するべく姿勢推定値を反復的に計算する。

別の態様は、姿勢推定値を反復的に計算するために速度スクリュウを計算する。

別の態様は、OpenGL、DirectX、レイトレーシングなど等のレンダリングアプリケーションプログラミングインタフェース(application programming interface、API)を用いて物体のモデル輪郭曲線を仮想画像に投影する。

【0008】

本発明のさらなる実施形態では、物体姿勢推定値は、推定姿勢をとる物体の画像、物体のモデル、および推定姿勢をとる物体の画像を撮るために用いられるカメラのパラメータを入力することと、2値マスク画像および画像深さ情報を得るべく、カメラのパラメータおよび初期姿勢パラメータを用いて物体のモデルを物体の仮想画像に投影することと、2値マスク画像および画像深さ情報を用いて初期姿勢パラメータを新しい姿勢パラメータに更新し、エネルギー関数を最小化するべくまたは反復の最大数に達するまで新しい姿勢パラメータを反復的に更新することと、によって精緻化される。

【0009】

本発明の別の態様は、コスト関数を用いて物体の内部および外部の平均値を計算することと、2値マスク画像から物体の輪郭Cを計算し、画像深さ情報を用いて3D輪郭点を計算することと、を含む。

【0010】

さらなる態様は、輪郭点についての、xおよびy方向における勾配 $T(x, y)$ を計算することと、画像ヤコビ行列を計算することと、ガウス・ニュートン法を用いて勾配流および速度スクリュウを計算することと、を含む。

【0011】

本発明の別の態様は、速度スクリュウおよび所与のステップワイズを用いて相対変位を計算することを含む。

【0012】

添付の図面と併せて以下の記載および請求項を参照することによって、本発明のより完全な理解とともに他の目的および達成が明らかになり理解されるようになる。

【図面の簡単な説明】

【0013】

【図1】図1は、本発明を利用するための物体処理デバイスおよびシステムの概略的なブロック図である。

【図2】図2は本発明の一般的方法のフローチャートである。

【図3】図3は、図2に示される方法の物体抽出段階のフローチャートである。

【図4】図4はピンホールカメラモデルにおける透視投影を示す。

【図5】図5はOpenGL投影および正規化デバイス座標(Normalized Device Coordinate、NDC)を示す。

【図6】図6(a)ないし(d)は、物体の境界に対する投影モデル曲線の位置の4つのすべてのあり得る場合を示す。

【図7】図7はカメラの運動の合成を示す。

【発明を実施するための形態】

10

20

30

40

50

【 0 0 1 4 】

以下の詳細な説明では、例示の目的で本発明の実施形態例を示す添付の図面に言及がなされる。図面において、いくつかの図全体にわたり、同じ符号は実質的に同種の構成要素を示す。これらの実施形態は、当業者が本発明を実施することを可能とするよう、十分詳しく記載されている。他の実施形態が利用されてもよく、構造的、論理的および電気的変更が、本発明の範囲から逸脱することなくなされてもよい。さらに、本発明の種々の実施形態は、異なるものであっても、必ずしも相互排他的であるわけではないことを理解されたい。例えば、1つの実施形態において記載されている特定の特徵、構造、または特性が他の実施形態内に含まれてもよい。従って、以下の詳細な説明は限定の意味で捉えられてはならず、本発明の範囲は、添付の請求項、およびこれらの請求項が権利を与えられる同等物の全範囲によってのみ定義される。

10

【 0 0 1 5 】

概して、実施形態例は、画像、とりわけ、1つ以上の物体のデジタル画像、を分析するための方法、デバイス、およびコンピューター可読媒体に関する。画像は、2D画像で表される対応する3D物体の姿勢（位置および配向）を推定するべく分析される。実施形態例は画像の2値マスクを作成し、2値マスクからシングレットのセットを抽出する。各シングレットは、物体の内側および外側輪郭内にある点を表すことになる。シングレットのセットは連結され、デュプレックス行列として表されるメッシュにされる。2つのデュプレックス行列が比較され、候補姿勢のセットを作る。その後、物体姿勢推定値が作られ、例えばロボットによる部品組み立てのための視覚システムにおける、後の利用のために記憶される。

20

【 0 0 1 6 】

図1に、物体処理デバイス100の例の略図が示されている。物体処理デバイス100は介在インターフェース102を介してホストコンピューター150とデータを交換する。ホストコンピューター150上にはアプリケーションプログラムおよび物体処理デバイスドライバーがアクセスのために記憶されてもよい。アプリケーションプログラムから画像取得コマンドが受信されると、例えば、物体処理デバイスドライバーはコマンドデータの、物体処理デバイス100に適したフォーマットへの変換を制御し、変換されたコマンドデータを物体処理デバイス100に送信する。ドライバーは同様に、物体処理デバイス100から種々の信号およびデータを受信して解釈し、ホストコンピューター150を介してユーザーに必要な情報を提供する。

30

【 0 0 1 7 】

ホストコンピューター150によってデータが送信されると、インターフェース102はデータを受信し、それを、RAM104の一部を形成する受信バッファー内に記憶する。RAM104は、例えばアドレス指定を通じて、多数の区画に分割され、受信バッファーまたは送信バッファー等の各種のバッファー群として割り当てられ得る。デジタル画像データ等のデータも物体処理デバイス100によって取り込み機構（単数または複数）112、フラッシュEEPROM110、またはROM108から得られ得る。取り込み機構（単数または複数）112は例えばカメラであり、プリンター等のデバイスの製造および/または組み立てにおいて用いられる部品等の1つ以上の物体を撮影することによってデジタル画像を生成することができる。カメラ112は、例えばロボット116、または人間によって制御され得るか、あるいは例えばコンピューター150によって自動制御され得る。次に、物体（単数または複数）のデジタル画像はRAM104の受信バッファーまたは送信バッファー内に記憶され得る。

40

【 0 0 1 8 】

プロセッサ106は、例えばROM108上またはフラッシュEEPROM110上に記憶されているコンピューター実行可能命令を用い、例えば方法200（図2）等の、或る機能または一群の機能を遂行する。方法200は本願明細書において以下でより詳細に説明される。RAM104の受信バッファー内のデータがデジタル画像である場合には、例えば、プロセッサ106はデジタル画像に対して方法200の方法手続きを実行し

50

、デジタル画像内の特徴を抽出し、抽出された特徴に基づき画像をさらに解析することができる。次に、画像が、例えばLCDディスプレイ等のディスプレイ114上に表示されたり、または例えば、プリンター162上での印刷、プロジェクター164を用いた投影、またはハードディスク160上への記憶のためにホストコンピューター150に転送されたりする前に、イメージングパイプラインにおけるさらなる処理がデジタル画像に対して遂行されてもよい。

【0019】

本願明細書において開示されている方法例200およびその変形は、非一時的コンピューター可読媒体であって、その上に記憶されたコンピューター実行可能命令またはデータ構造を運ぶまたは有するための媒体を用いて実装され得る。このようなコンピューター可読媒体は、汎用または専用コンピューターのプロセッサによってアクセスされ得る任意の利用可能な媒体であることができる。例として、ただし限定ではなく、このようなコンピューター可読媒体は、RAM、ROM、EEPROM、CD-ROMまたは他の光ディスクストレージ、磁気ディスクストレージまたは他の磁気記憶デバイス、あるいはコンピューター実行可能命令またはデータ構造の形式でプログラムコードを運ぶまたは記憶するために用いられ得るとともに、汎用または専用コンピューターのプロセッサによってアクセスされ得る任意の他の媒体を含むことができる。以上のものの組み合わせもコンピューター可読媒体の範囲に含まれるべきである。

10

【0020】

コンピューター実行可能命令は、例えば、汎用コンピューターまたは専用コンピューターのプロセッサに或る機能または一群の機能を遂行させる命令およびデータを含む。本願明細書においては、主題は方法手続きに固有の言葉で記載されているが、添付の請求項において定義される主題は、本願明細書において記載されている特定の手続きに必ずしも限定されるものではないことを理解されたい。むしろ、本願明細書において記載されている特定の手続きは、請求項を実装する形態の例として開示されている。

20

【0021】

専用コンピューターの例としては、デジタルカメラ（その例としては、日本、長野県諏訪市大和に本社を置くセイコーエプソン社製のエプソンR-D1デジタルカメラが挙げられるが、それに限定されるものではない）、デジタルカムコーダー、プロジェクター、プリンター、スキャナー、複写機、ポータブルフォトビューアー（その例としてはセイコーエプソン社製のエプソンP-3000もしくはP-5000ポータブルフォトビューアーが挙げられるが、それらに限定されるものではない）、またはポータブルムービープレイヤー、あるいはプリンター/スキャナー/複写機の組み合わせ（その例としてはセイコーエプソン社製のエプソンスタイラスフォトRX580、RX595、もしくはRX680、エプソンスタイラスCX4400、CX7400、CX8400、もしくはCX9400Fax、およびエプソンAcuLaser（登録商標）CX11NFが挙げられるが、それらに限定されるものではない）またはプリンター/スキャナーの組み合わせ（その例としてはすべてセイコーエプソン社製のエプソンTM-J9000、TM-J9100、TM-J7000、TM-J7100、およびTM-H6000IIIが挙げられるが、それらに限定されるものではない）またはデジタルカメラ/カムコーダーの組み合わせ等の、それらの何らかの組み合わせ等の画像処理デバイスが挙げられる。

30

40

【0022】

入力画像の仕様

図2を参照すると、方法200における最初のステップは画像を入力することである（ステップ210）。入力画像は、カメラ（取り込み機構）112を操作するロボット116によって生成され得る。入力画像が1つの物体を包含しているならば、このとき、それは好ましくは画像エリアの少なくとも10%を占めなければならない。画面内に1つを超える物体があるならば、対象となっている物体は好ましくは画像（画面）内の他のいずれの物体よりも少なくとも2倍は大きくなければならない。入力画像は、例えば、RAM104またはハードディスク160内に記憶されている物体のCADモデルであることもで

50

きる。好ましい実施形態では、以下において記載されるように、物体の複数の画像が入力され、解析される。複数の画像の各々は好ましくは、他の画像の各々に示される視像とは異なる、物体の視像を包含する。これらの複数の画像は、C A Dモデル、および/またはカメラを操作するロボット116を用いて生成され得る。

【0023】

物体位置特定および抽出

本発明の物体位置特定(ステップ212)および物体抽出(ステップ214)段階は、画像または画面内に存在する物体を検出し、そのバウンディングボックスを計算する。物体抽出段階は物体の内側輪郭および外側輪郭の両方を抽出する。これらのステップ/段階は図3においてより詳細に示される。

10

【0024】

一般的に言えば、これらのステップは、無地の背景上に1つ以上の物体を包含する単一チャンネル画像310(図3)を処理する。これらのステップは基本的に画像内の活動のエリアを発見するものであり、従って、背景内のアーチファクトに敏感であってよい。入力画像内の物体は好ましくは単色であり且つ背景と物体との間の最小限のコントラストを有する。背景の色または異なる物体の色に対する追加の制限はない。物体のうちのあるものは背景よりも暗くてよく、あるものはそれよりも明るくてよい。物体は同様の色を有してもよいしまたは異なる色を有してもよい。

【0025】

活動計算(図3、ステップ312)

20

活動計算ステップの間、画像310はサイズ $S \times S$ のブロックに分割される。 S は画素の数である。例として、 S は8である。ただし、特定の環境に従って、異なる値が選択され得る。この設定パラメータ S は本願明細書においてACTIVITY・SCALEと呼ばれる。次に、本ステップは各ブロックについての標準偏差および平均強度を計算し、2つの行列、標準(std)および平均(ave)、として出力を返す。

【0026】

活動的エリア選択(ステップ314)

本ステップでは、本発明は行列stdの要素を閾値と比較し、閾値を超える要素を特定する。閾値は特定の環境のために選択され得る。例えば、これは色の閾値であり得よう。色深度を表すために8ビットが用いられるならば、このとき、閾値は0と255との間のどこかに設定されよう。理解されるように、活動的エリア選択ステップは、無地の背景と対照的にカラー物体に対応する画素ブロックを特定する。選択された活動閾値を超えると特定された要素は活動マップに入れられる。

30

【0027】

領域解析(ステップ316)

領域解析ステップ316は、前のステップ314において作られた活動マップに対するプロブ解析を遂行する。本ステップにおいて用いられるパラメータは、OBJECTS・SIZE・MIN(デフォルト値は0.001)およびOBJECTS・SIZE・MAX(デフォルト値は0.1)によって制御されるサイズ、ならびに候補領域の数OBJECTS・COUNT(デフォルト値は8)である。これらの設定パラメータを画素からブロックに変換するために、サイズ制限はまずACTIVITY・SCALEの2乗で除される。領域解析ステップは各物体のためのバウンディングボックス、ならびにその1次および2次平均および面積を返す。

40

【0028】

領域解析ステップの間、各領域はインデックスを与えられる。領域のインデックスは1から始まるが、必ずしも連続したものとなるわけではない。換言すると、サイズ制約に一致しなかった領域に対応する、欠けているインデックス値がある。インデックスマップは、各ブロックに付与されたインデックスを保持する。このマップは境界マップと呼ばれる。境界マップ内の、非ゼロのインデックスが与えられたブロックはいずれも境界ブロックである。残りのブロックは背景であるかまたは物体の内部を表すかのいずれかである。内

50

部ブロックは、本願明細書において後に記載されるステップ 3 2 0 において検出される。

【 0 0 2 9 】

閾値計算 (ステップ 3 1 8)

閾値計算ステップ 3 1 8 は、各領域のための境界ブロックの内容を前景と背景とに分けるための最適閾値を求める。この手順は、領域毎に個別になされる以下の段階を通じて実行される。ここで、 I は、領域のための境界ブロックのうちの一つに入るすべての点のセットである。

【 0 0 3 0 】

- 1 . I の内容についての 2 5 6 ビンヒストグラムを作る。
- 2 . I の平均として t を計算する。
- 3 . t の値を用いて I を閾値処理し、 I^+ および I^- を作る。
- 4 . I^+ の平均として t^+ を計算する。
- 5 . I^- の平均として t^- を計算する。
- 6 . $(t^+ + t^-) / 2$ として t^* を計算する。
- 7 . t と t^* との間の差が 1 未満であれば、このときは、 t を閾値として返し、さもなければ、 $t = t^*$ と設定し、3 に行く。

本段階において計算された閾値 (`OSMX_THRESHOLD`) は後のステップにおいて用いられる。

【 0 0 3 1 】

閾値計算ステップ 3 1 8 は、物体は背景よりも暗いのかまたはその逆であるのかも判定する。この判定は、各物体のためのバウンディングボックスの境界にある画素を調べ、閾値を超えるそれらの数を求めることに基づく。この数が、集計された画素の総数の半分以上を超えるならば、このとき、物体は背景よりも暗いと判定される。この情報 `OSMX_INVERSE` は後のステップにおいて用いられる。例えば、`OSMX_INVERSE` は、物体が背景よりも暗ければ 1 に、明るければ 0 に設定され得よう。

【 0 0 3 2 】

内部検出 (3 2 0)

内部検出ステップ 3 2 0 は境界マップを用い、内部マップを構築する。この手順は、すべてゼロの内部マップより開始し、領域毎に、そのバウンディングボックス内の各ブロックについて以下の操作を遂行する。

- 1 . ブロックが境界マップ内になく且つ、ブロックについての平均が、(`OE_INVERSE` がどのように設定されているかに依存して) `OE_THRESHOLD` を上回りまたは下回り且つ、このブロックが境界マップ内に少なくとも一つの 4 連結隣接ブロックを有するならば、このときは、このブロックを内部マップ内に標識する。

【 0 0 3 3 】

境界閉塞 (ステップ 3 2 2)

境界マップが内部マップを包囲しないということがあり得る。その結果、次に記載される物体抽出段階の性質により、抽出された物体内に、望ましくない「くぼみ」が出現する可能性がある。この問題を解決するために、境界閉塞ステップ 3 2 2 は、各領域のバウンディングボックス内で以下の操作を遂行することによって各領域のための境界マップを閉塞する。

- 1 . ブロックが境界マップ内にもまたは内部マップ内にもなく且つ、それが内部マップ内に少なくとも一つの 4 連結隣接ブロックを有するならば、このときは、このブロックを境界マップ内に標識する。

【 0 0 3 4 】

物体抽出 (ステップ 3 2 4)

物体抽出ステップ 3 2 4 は、境界マップおよび内部マップに基づき物体を抽出する。この手順は各物体についてそれ自身のバウンディングボックス内で以下の操作を遂行する。

- 1 . このブロックが内部マップ内に標識されているならば、このときは、それに対応するすべての画素を物体に属するものとして設定する。

10

20

30

40

50

2. このブロックが境界マップ内に標識されているならば、このときは、ブロックについての平均が、(OE__INVERSEがどのように設定されているかに依存して)OE__THRESHOLDを上回るまたは下回るならば、それに対応するすべての画素を物体に属するものとして設定する。

【0035】

デュプレット作成 (図2、ステップ216)

デュプレット作成は、実際の画像が用いられる、本発明における最後のステップである。本段階の後では、作業はデュプレット行列ならびに他の特徴点およびベクトルに対してなされる。

【0036】

デュプレット作成は、以下に各々記載される、輪郭追跡、シングレット検出、およびデュプレット作成の3つのサブブロックを包含する。

【0037】

輪郭追跡

輪郭追跡は、物体抽出段階によって返されたすべての輪郭を追跡する手順である。本段階において収集される情報はシングレット検出手順において用いられる。本発明においては、特定のアルゴリズムに限定されない任意の周知の輪郭追跡アルゴリズムが用いられ得る。利用されてよい輪郭追跡アルゴリズムの例としては矩形追跡、ムーア近傍、放射状走査が挙げられる。好ましい実施形態では、Theo Pavlidisのアルゴリズムが用いられる。

【0038】

シングレット検出

本段階は、特定の環境のために選択され得る2つのパラメータTRACE__WINDOWおよびSINGLETT__DISTANCEに依存する。これらの両パラメータのためのデフォルト値は12画素である。TRACE__WINDOWの2倍よりも長いすべての輪郭について、以下の手順が輪郭上の点毎に実行される。

1. 輪郭上の点の周りのTRACE__WINDOWの点についての平均のXおよびY座標を求める。
2. 輪郭上の点を、TRACE__WINDOWだけ左および右に離れた2つの点に連結する。
3. 2つの角度を求め、それらを入射角度および出射角度と呼ぶ。
4. 入射角度および出射角度を表す2つのベクトルの内積を求める。この内積の符号を変更し、それを曲率と呼ぶ。
5. この点とその周りのSINGLETT__DISTANCEの点群内の最大曲率を有するならば、その点をシングレットとして標識する。
6. 各シングレットについてのXおよびY座標を、入射および出射角度ならびにそれらに対応する曲率とともに記憶する。

【0039】

デュプレット作成

デュプレット作成は、検出されたシングレットを組み合わせるデュプレットにする手順である。デュプレットを作成するプロセスは以下のステップを包含する。

1. その(i, j)要素がi番目とj番目のシングレットの間の距離を包含する行列Dを作る。
2. 最小閾値未満であるかまたは最大閾値を上回るDの要素をすべてNAN(Not a Number (非数)、初期化されていない変数を示す特定記号)として変更する。最小閾値および最大閾値は、DISTANCE_MIN(デフォルト値は0.01)およびDISTANCE_MAX(デフォルト値は0.25)に画像の最小寸法を乗じられたものとしてそれぞれ計算される。
3. すべてゼロのS x 1配列を作り、それをdegと呼ぶ。deg(s)はs番目のシングレットに連結されるシングレットの数を包含することになる。

10

20

30

40

50

4. DがNANしか包含しなくなるか、またはシングレットのための最大数が指定されている場合には該最大に達するまで、Dをスキャンする。

5. Dの最小の要素を見いだす。それを (s_1, s_2) と呼ぶ。 $deg(s_1)$ および $deg(s_2)$ の両方をインクリメントする。 $deg(s_1)$ または $deg(s_2)$ のどちらかが最大度数DEGREE(デフォルトは10)よりも大きくなれば、このときは、該シングレットに対応する行および列内のすべての要素をNANとして設定する。例えば、 $deg(s_1)$ がDEGREEよりも大きくなれば、このときは、すべてのsについて $D(s, s_1)$ および $D(s_1, s)$ をNANに設定する。

6. 対 (s_1, s_2) について、それらを一緒に連結する角度ならびにそれらを連結する線の長さを計算し、それらをそれぞれ角度および距離と呼ぶ。両シングレットの一時的複製の入射および出射角度から角度を減じる。これで、デュプレットは \langle 角度, 距離, s_1 .入射, s_1 .出射, s_2 .入射, s_2 .出射 \rangle として表されることになる。

7. デュプレット行列を返す。

【0040】

デュプレット比較(図2、ステップ218)

デュプレット比較は、2つのデュプレット行列内の項目を比較し、一致のリスト、および従って候補姿勢の第1のリストを作るプロセスである。2つのデュプレット行列はD1およびD2であるとして、プロセスは以下のように遂行される。

【0041】

1. D1内のあらゆるデュプレットd1およびD2内のd2について、これらの条件がすべて満足されているかどうかを判定する。

$$|d_1.s_1.入射 - d_2.s_1.入射| < THRESHOLD$$

$$|d_1.s_1.出射 - d_2.s_1.出射| < THRESHOLD$$

$$|d_1.s_2.入射 - d_2.s_2.入射| < THRESHOLD$$

$$|d_1.s_2.出射 - d_2.s_2.出射| < THRESHOLD$$

ここで、THRESHOLDは設定変数DUPLETS.THRESHOLDによって決められる。

これらの条件が満たされれば、このときは、候補姿勢(d1.長さ/d2.長さ, d1.角度-d2.角度)を候補のリストに追加する。さらに、2つのシングレットおよび両デュプレットのXおよびY座標に注目する。この候補を「直」としてタグ付けする。

【0042】

2. 同様に、これらの条件が満足されるかどうかを判定する。

$$|d_1.s_1.入射 - d_2.s_2.入射| < THRESHOLD$$

$$|d_1.s_1.出射 - d_2.s_2.出射| < THRESHOLD$$

$$|d_1.s_2.入射 - d_2.s_1.入射| < THRESHOLD$$

$$|d_1.s_2.出射 - d_2.s_1.出射| < THRESHOLD$$

この場合、d1.角度-d2.角度-パイと記録されることになる角度を除き、同様の候補姿勢がリストに追加されることになる。この候補は「逆」としてタグ付けされることになる。

【0043】

3. 候補のリストを、次に記載されるクラスター化段階に渡す。

候補クラスター化(図2、ステップ220)

候補クラスター化は、デュプレット比較を通じて作られた姿勢候補を取り入れ、候補の大きなセットによって各々表される限られた数の姿勢を推定するプロセスである。本手順は、いずれも以下において記載される2つの段階、スケール-角度クラスター化、およびX-Yクラスター化、を包含する。

【0044】

スケール-角度クラスター化

本段階の目的は、デュプレット比較を通じて作る候補姿勢に基づき候補(スケール, 角度)対のリストを作ることである。

10

20

30

40

50

S および A は、候補スケールおよび角度を持つ 2 つの $1 \times N$ であるとする。スケール - 角度クラスター化手順は以下のようになされる。

1. S の値の底 B A S E の対数を計算することによって S _ L を作る。
2. A および S _ L のヒストグラムを独立に作る。各ヒストグラムは B I N S のピンを包含することになる。A についてのヒストグラムは - パイないしパイのレンジに及ぶことになる。S _ L は、S C A L E . M I N および S C A L E . M A X のレンジ内で、どちらも底 B A S E で解析されることになる。
3. どちらのヒストグラム上の点も、もしそれがその右側の D の点よりも高く且つその左側の D の点よりも高いかまたはそれに等しければ、「ピーク」として標識する。
4. 各ヒストグラムについて C 1 個の最高ピークを独立に選択する。
5. すべてのスケール候補をすべての角度候補と交差させ、スケール - 角度候補のリストを作る。このリストを X - Y クラスター化段階に渡す。各スケール - 角度候補について、この対を生じさせた一致デュプレット内のシングレットの位置も送る。
6. 各スケール - 角度候補についての信頼性を、それを表すデュプレット一致の対の数を 2 つのデュプレット行列内のデュプレットの数の最小で除したものとして計算する。
7. スケール - 角度候補の数が C 2 を超えるならば、このときは、最高の信頼性を有する C 2 個のものを取る。

【 0 0 4 5 】

X - Y クラスター化

スケール - 角度クラスター化段階の出力は (スケール, 角度) 候補のセットである。X - Y クラスター化段階は、各 (スケール, 角度) 候補について以下のように、候補に平行移動情報を加えるプロセスである。

1. この候補についてのシングレット位置を、軸系の中心が画像の中心に移動するように変更する。このプロセスは 2 つのデュプレット行列について独立に実行されることになる。
2. クエリシングレット位置を角度によって回転させる。
3. クエリシングレットをスケールによってスケールリングする。
4. クエリと参照との間のシングレット位置についての X および Y における差を独立に求める。これらを d_x および d_y 配列と呼ぶ。
5. 本段階においては、 $d_x - d_y$ 対は、スケール - 角度対に対して実行されるものと同様の 2 次元クラスター化アルゴリズムを受けられることができるか、またはより迅速な演算が続いて行われることができる。
6. d_x および d_y の平均を独立に求め、この情報をスケール - 角度対に付加し、かくして、姿勢候補 (角度, スケール, d_x , d_y , 信頼性) を作る。この姿勢候補についての信頼性はスケール - 角度対についての信頼性に等しいままとすることになる。
7. 設定パラメーターによって要求される場合には、以下において記載されるように、重複調査を遂行する。
8. 信頼性の最高値に対応する C 2 個の候補を取り、それらを次の段階に報告する。

【 0 0 4 6 】

重複の考慮

重複の考慮は O V E R L A P が非負である場合のみ機能する。重複の考慮は以下のようになされる。

1. (角度, スケール, d_x , d_y , 信頼性) 情報に従い参照バウンディングボックスを変換する。回転され、スケールされ、移動されたバウンディングボックスを覆う最小の矩形を見いだす。
2. 変更された、参照のバウンディングボックス、およびクエリのバウンディングボックスを両方とも覆う最小の矩形を見いだす。この矩形の面積を A と呼ぶ。
3. もし、 $\min(A_1, A_2) / A$ が O V E R L A P 未満であれば、この候補を不適当として標識する。ここで、 A_1 および A_2 は、参照の変更されたバウンディングボックスの面積、およびクエリの面積である。

10

20

30

40

50

4. 不相当として標識された候補をリストから削除する。

【0047】

輪郭デュプレットデータベース(図2、222)

輪郭デュプレットデータベース(Contour-Duplets Database、CDDDB)ファイルは、各々仰角および方位角に付随するデュプレット行列のリストを包含する。CDDDBファイルを生成するために、上述のアルゴリズムは訓練用画像(図2、224)に対してデュプレット作成を遂行することになり、次に、結果として生じる、仰角-方位角表示を各々伴うデュプレット行列をCDDDB222内に記憶することになる。これらのサンプルは視像と呼ばれる。

CDDDBファイルは実際には、訓練用画像に関する詳細を包含するファイルを伴う。これらの詳細は様々な視覚化を作るために必要なものであり、アルゴリズムの通常の演算には必要とされない。

クエリが実行されると、手順はデータベース内の視像毎に繰り返される。アルゴリズムの最終出力は、最高の対応する信頼性値を有する、CDDBC.SEARCH.CANDIDATES.TOTALの数の候補である。次に、各候補を作った視像の角度表示が各候補に付加され、かくして姿勢の(仰角、方位角、角度、スケール、dx、dy、信頼性)表示を作る。以上に概説された物体検出ステップの結果は、姿勢推定値である、姿勢のこれらの表示であり、メモリー、例えばRAM104またはハードディスク160、内に記憶される。これらの姿勢推定値は次に、製品の組み立ての際の物体/部品のロボット認識において用いられ得る。

【0048】

出力の視覚化

様々な視覚化が作られ得る。これらの視覚化の2つがモデルビューおよびオーバーレイである。モデルビューとは、スケール、角度、および平行移動も反映するように変更された一致訓練用画像の画像である。オーバーレイとは、クエリ画像上のモデルビューのオーバーレイである。

【0049】

物体モデルの仕様

現在、訓練手順は、様々なパラメーターによって支配される通りの、様々な姿勢をとる対象となっている物体を示す画像を必要とする。これらの画像を作る実用的且つより簡便な方法が、物体の3D視像を作る3次元視覚化ツールであるサンプルシミュレーター228を用いるものである。

【0050】

設定ユーティリティ(図2、230)

上述されたように、本発明の方法の異なる部分は異なる設定パラメーターを用いる。これらのパラメーターのいくつかの例示的な(デフォルト)値が上述の記載において与えられている。しかし、これらの値はユーザーによって特定の環境に合うように設定され得る。設定ユーティリティは、これらのパラメーターのための適切な値を求め、それにより、ユーザーからの直感的入力を用いるのは最小限にしてそれらを設定する対話型ツールである。

【0051】

以下のセクションは、特に物体の画像がカメラ画像である場合に物体の姿勢推定値を精緻化する本発明の態様を記載する。

【0052】

本発明の姿勢精緻化はアラインメント問題としてモデル化される。我々は、対象となっている物体、すなわちロボットが特定し、拾い上げなければならない部品、の3Dモデル、および物体を撮像するために用いられるカメラ112のカメラパラメーターより開始する。目標は、モデルに対してそれが画像内の対象となっている物体と合致するように適用されるべき3D(ユークリッド、6自由度(6 degrees of freedom))変換を見いだすことである。その結果、姿勢精緻化はコスト関数の反復的な最適化問

10

20

30

40

50

題として実装される。物体前景を負担するコスト以外に、コスト関数は、局所の特徴点または境界特徴点を合わせるまたは一致させる画像一致コストを含んでよい。

【0053】

アルゴリズムをより良く理解するために、コスト関数および緩和解の定義の前にカメラ投影モデル、OpenGL仮想カメラモデルおよび画像ヤコビ行列が導出される。

【0054】

カメラ投影モデル

本発明のカメラ投影モデルはピンホールカメラモデルに基づく。ピンホールカメラはカメラ機能の最も単純で理想的なモデルである。それは無限小の穴を有し、そこを通過して光が入り、該穴と向かい合うカメラ表面上に倒立像を形成する。図4に示されるように、透視投影は3次元物体の、2次元画像平面上への投影である。実際には、画像の焦点をカメラの焦点面上に合わせるためにレンズを用いなければならない。焦点面は、画像平面に平行である1つの特定の平面である。ピンホールカメラ座標はワールド座標に合わせられるとすると、単純な幾何学は以下を示す。

$$\frac{x_{im}}{f} = \frac{x_w}{z_w} \quad \text{および} \quad \frac{y_{im}}{f} = \frac{y_w}{z_w} \quad (1)$$

ここで、 (x_w, y_w, z_w) は3Dワールド座標における物体の点であり、 (x_{im}, y_{im}) は2D画像平面座標における点である。

【0055】

カメラフレーム座標から画素(画像)座標への投影モデルは以下のように導出される。

$$CP = \begin{bmatrix} \frac{F}{S_x} & 0 & 0 \\ 0 & \frac{F}{S_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2) \text{ とする。}$$

ここで、 F は焦点距離であり、 S_x および S_y は、焦点にあるセンサー平面から画像または画素座標への、 x および y 方向におけるスケーリング係数である。 S_x および S_y スケーリング係数はCCDセンサーのセルの幅および高さ、ならびに分解能に比例する。我々は、モデルスケーリング係数 λ_x および λ_y を得るようにカメラを較正することができる。

$$x_{im} - x_0 = \lambda_x \frac{x_c}{z_c} \quad \text{および} \quad y_{im} - y_0 = \lambda_y \frac{y_c}{z_c} \quad (3)$$

ここで、 (x_0, y_0) は画像または画素座標におけるカメラ座標の原点の投影である。

【0056】

$u = x_{im} - x_0$ および $v = y_{im} - y_0$ とすると、式(3)は以下のように表され得る。

$$u = \lambda_x \frac{x_c}{z_c} \quad \text{および} \quad v = \lambda_y \frac{y_c}{z_c} \quad (4)$$

あるいは

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = CP * \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (5)$$

10

20

30

40

50

【 0 0 5 7 】

画像ヤコビ行列

画像ヤコビ行列は、作業空間内の或る物体の速度と、作業空間の観察される画像内で起きる対応する変化との間の関係に基づいて構築されなければならない。

【 0 0 5 8 】

特徴点 S の現在の投影は座標 $f = (u, v)$ を有する。画像ヤコビ行列は以下のように算定される。

$$\begin{aligned} \dot{f} &= J(r)\dot{r} \\ \dot{r} &= [V; \Omega]^T = [v_x, v_y, v_z, w_x, w_y, w_z]^T \\ J(r) &= \left[\frac{\partial f}{\partial r} \right] = \begin{bmatrix} \frac{\partial f_1(r)}{\partial r_1} & \dots & \frac{\partial f_1(r)}{\partial r_m} \\ \dots & \dots & \dots \\ \frac{\partial f_k(r)}{\partial r_1} & \dots & \frac{\partial f_k(r)}{\partial r_m} \end{bmatrix} \end{aligned} \quad 10$$

【 0 0 5 9 】

座標 (x_c, y_c, z_c) またはカメラフレームにおける (x, y, z) を有する点 P を考える。透視投影を用いると、 P の画像平面座標は式 (4) によって与えられる。

20

$$u = \lambda_x \frac{x}{z} \quad \text{および} \quad v = \lambda_y \frac{y}{z} \quad (6)$$

【 0 0 6 0 】

カメラフレームにおける座標 (x, y, z) を有する単一の点についての画像ヤコビアンを構築するために、式は以下のステップで導出される。

【 0 0 6 1 】

u および v についての時間導関数を算出する。

30

これらを u 、 v 、 \dot{x} 、 \dot{y} 、ならびに \dot{z} および z に関して表現する。

\dot{x} 、 \dot{y} 、および \dot{z} のための、制御入力 (速度スクリュー)

$$\dot{r} = [V; \Omega]^T = [v_x, v_y, v_z, w_x, w_y, w_z]^T \quad \text{および} \quad x, y, z \text{ に関する表現式を求める。}$$

式を組み合わせ、代数を解く。

【 0 0 6 2 】

ステップ 1、割り算の規則を用いると、

$$\begin{aligned} \dot{u} &= \lambda_x \left(\frac{\dot{x}}{z} - \frac{x\dot{z}}{z^2} \right) = \lambda_x \frac{\dot{x}z - x\dot{z}}{z^2} \\ \dot{v} &= \lambda_y \left(\frac{\dot{y}}{z} - \frac{y\dot{z}}{z^2} \right) = \lambda_y \frac{\dot{y}z - y\dot{z}}{z^2} \end{aligned} \quad (7) \quad 40$$

【 0 0 6 3 】

ステップ 2、透視投影の式は、以下のように x および y のための表現式を与えるように書き直され得る。

$$x = \frac{uz}{\lambda_x} \quad \text{および} \quad y = \frac{vz}{\lambda_y} \quad (8)$$

これらを式 (7) に代入する。

$$\begin{aligned} \dot{u} &= \lambda_x \left(\frac{\dot{x}}{z} - \frac{x\dot{z}}{z^2} \right) = \lambda_x \frac{\dot{x}}{z} - \frac{u\dot{z}}{z} \\ \dot{v} &= \lambda_y \left(\frac{\dot{y}}{z} - \frac{y\dot{z}}{z^2} \right) = \lambda_y \frac{\dot{y}}{z} - \frac{v\dot{z}}{z} \end{aligned} \quad (9)$$

10

【 0 0 6 4 】

ステップ 3、カメラフレームに対する (固定点) P の速度は次式によって与えられ

$$\dot{P} = -\Omega \times P - V$$

または

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = - \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} - \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (10)$$

20

ここで、速度スクリュー $\dot{r} = [V; \Omega]^T = [v_x, v_y, v_z, w_x, w_y, w_z]^T$ である

式 (10) を展開すると

$$\begin{aligned} \dot{x} &= -v_x - w_y z + w_z y \\ \dot{y} &= -v_y - w_z x + w_x z \\ \dot{z} &= -v_z - w_x y + w_y x \end{aligned} \quad (11)$$

【 0 0 6 5 】

ステップ 4、式を組み合わせ、次式を得る

$$\begin{aligned} \dot{u} &= -\frac{\lambda_x}{z} v_x + \frac{u}{z} v_z + \frac{uv}{\lambda_y} w_x - \frac{\lambda_x^2 + u^2}{\lambda_x} w_y + \frac{\lambda_x v}{\lambda_y} w_z \\ \dot{v} &= -\frac{\lambda_y}{z} v_y + \frac{v}{z} v_z + \frac{\lambda_y^2 + v^2}{\lambda_y} w_x - \frac{uv}{\lambda_y} w_y - \frac{\lambda_y u}{\lambda_x} w_z \end{aligned}$$

30

これらの式は画像ヤコビ行列形式で書かれることができる。

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = J(r) \dot{r} = \begin{bmatrix} -\frac{\lambda_x}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda_y} & -\frac{\lambda_x^2 + u^2}{\lambda_x} & \frac{\lambda_x v}{\lambda_y} \\ 0 & -\frac{\lambda_y}{z} & \frac{v}{z} & \frac{\lambda_y^2 + v^2}{\lambda_y} & -\frac{uv}{\lambda_y} & -\frac{\lambda_y u}{\lambda_x} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \quad (12)$$

40

【 0 0 6 6 】

OpenGL 仮想カメラモデル

モデルを画像上に投影するために、本発明は好ましくは、既存のハードウェアを用いて

50

高速レンダリングおよび隠線消去を実行することができるOpenGLを用いる。加えて、OpenGLは各画素における深さ情報を記憶することができる。この情報を用いて、各輪郭点に対応する3D点は計算される。図5はOpenGL投影および正規化デバイス座標(NDC)を示す。好ましい実施形態ではOpenGLが用いられるが、本発明はこのレンダリングアプリケーションプログラミングインタフェース(API)に限定されるものではなく、DirectX、レイトレーシングなど他のレンダリングAPIが利用されてもよい。

【0067】

透視投影では、角錐台(視点座標)における3D点は立方体(NDC)に写像され、x座標は[1, r]から[-1, 1]に、y座標は[b, t]から[-1, 1]に、z座標は[n, f]から[-1, 1]に写像される。

【0068】

投影行列は視点座標系における頂点をクリップ座標系に変換する。

$$\begin{pmatrix} X_{clip} \\ Y_{clip} \\ Z_{clip} \\ W_{clip} \end{pmatrix} = \begin{pmatrix} \cot(fovy/2) & 0 & 0 & 0 \\ 0 & \cot(fovy/2) & 0 & 0 \\ 0 & 0 & \frac{Far+Near}{Near-Far} & \frac{2*Far*Near}{Near-Far} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{pmatrix} \quad (13)$$

【0069】

スケーリング行列および画素座標計算を適用すれば、OpenGLにおける3D頂点の新しい投影は以下ようになる。

$$\begin{aligned} u_{virtual} &= \frac{\cot(fovy/2)}{aspect} \frac{width}{2} \left(\frac{X_e}{Z_e} \right) + \left(\frac{width}{2} + x_0 \right) \\ v_{virtual} &= -\cot(fovy/2) \frac{height}{2} \left(\frac{Y_e}{Z_e} \right) + \left(\frac{height}{2} + y_0 \right) \end{aligned} \quad (14)$$

式(3)および(14)を用いると、OpenGL仮想カメラの、実カメラモデルへの対応を求めることができる。

【0070】

カメラ座標およびOpenGL仮想カメラの視点座標の定義は異なることに留意されたい。その結果、2つの座標の間の対応を導出するには以下の変換が必要となる。

$$\begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{pmatrix} \quad (15)$$

本発明では、OpenGLは、モデルを投影するためにのみ用いられる。すべての式はデフォルトでは実カメラモデルを用いて導出される。

【0071】

2.4 コスト関数

本発明の姿勢精緻化方法はエネルギーベースのコスト関数を最小化する。コスト関数は、3D姿勢(6DoF)を用いるモデル投影が画像内の対象となっている物体と合致するよう、最小値を有するように定義される。

$$E = F_1 + F_2 = \int T(x, y) * (I(x, y) - u_1)^2 dx dy + \int (1 - T(x, y)) * (I(x, y) - u_2)^2 dx dy \quad (16)$$

ここで、 $I(x, y)$ は入力画像であり、 $T(x, y)$ は所与の 3D 姿勢を有するモデル投影の 2 値マスク画像である。

【0072】

2 値マスクの $T(x, y)$ の画像は所与の 3D 姿勢およびカメラパラメータについてのモデルの投影であり、以下の通りの定義を有する。

$$T(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ inside} \\ 0 & \text{outside} \end{cases} \quad (17)$$

10

OpenGL は、物体モデルを 2D 仮想画像に投影することによって 2 値マスク画像の高速レンダリングを実行することができる。

【0073】

u_1 および u_2 は物体マスクの内部および外部の入力画像の平均値であり、以下のように定義される。

$$u_1 = \frac{\int T(x, y) * I(x, y) dx dy}{\int T(x, y) dx dy}$$

$$u_2 = \frac{\int (1 - T(x, y)) * I(x, y) dx dy}{\int (1 - T(x, y)) dx dy} \quad (18)$$

20

【0074】

投影モデル輪郭曲線 C は、画像を物体の「内側」と物体の「外側」とに分ける $T(x, y)$ から得られ得る。項 F_1 は物体の内部のフィッティングエネルギーであり、 F_2 は物体の外部のフィッティングエネルギーである。

【0075】

単純な場合のコスト関数の基本的な考え方は以下のように説明される。画像 $I(x, y)$ は、物体および背景についての u_{obj} および u_{back} の個別の値の、おおよそ区分的に一定な強度の 2 つの領域によって形成されるとする。画像 $I(x, y)$ 内の物体の境界は C_0 によって示される。投影モデル輪郭曲線 C が画像 $I(x, y)$ 内の物体境界 C_0 と合致するならば、このとき、物体の内部では $u_1 = u_{obj}$ 、物体の外部では $u_2 = u_{back}$ を得、その結果、次式を得ることができる。

$$\inf \{ F_1(C) + F_2(C) \} = 0 \quad F_1(C_0) + F_2(C_0)$$

【0076】

この場合、投影モデル輪郭曲線 C が画像内の物体境界 C_0 と合致するとエネルギー関数は最小化されることは明らかである。

40

【0077】

これは、図 6 に示されている通りの単純な場合において理解され得る。同図は投影モデル曲線の位置のあり得る場合をすべて示している。例えば、図 6 (a) のように曲線 C が物体の外部にある場合には、 $F_1(C) > 0$ 且つ $F_2(C) = 0$ である。図 6 (b) のように曲線 C が物体の内部にある場合には、このときは、 $F_1(C) = 0$ 且つ $F_2(C) > 0$ である。図 6 (c) のように曲線 C が物体の内部および外部のどちらにもある場合には、このときは、 $F_1(C) > 0$ 且つ $F_2(C) > 0$ である。最後に、図 6 (d) に示されるように $C = C_0$ で投影モデル曲線 C が物体の境界上にある場合には、 $F_1(C) = 0$ 且つ $F_2(C) = 0$ であり、フィッティングエネルギーは最小化される。

【0078】

50

我々のコスト関数の定義に基づき、例えばRAM 104またはハードディスク160内に記憶されている姿勢推定値は、上述のフィッティング項を最小化するべく更新されることになり、我々は、一致した特徴点または境界特徴からいくつかの他の項または制約を加えることができる。例えば、もし、2D入力画像 $p_j = (p_{jx}, p_{jy})^T$ と、仮想画像内に $(P_j) = (u_j, v_j)^T$ として投影される3Dモデル特徴 $P_j = (X_j^{obj}, Y_j^{obj}, Z_j^{obj})$ との間に一致した特徴点のセット（例えば $j = 1 \dots N$ 個の一致対）があれば、追加の制約項は以下のように定義されてよい。

$$E_{match} = \sum_{j=1}^N w_j * \|\pi(P_j) - p_j\| = \sum_{j=1}^M w_i * ((u_j - p_{jx})^2 + (v_j - p_{jy})^2) \quad (19) \quad 10$$

ここで、 w_j は特徴一致についての重みであり、それは一致信頼性またはスコアであってよく、 $(P_j) = (u_j, v_j)^T$ はモデル特徴点の、 $P_i = (X_i^{obj}, Y_i^{obj}, Z_i^{obj})$ から所与の3D姿勢を有する画像への投影を示すことである。は画像座標におけるユークリッド距離であってよい。

【0079】

式(19)すなわち一致エネルギーの最小化は精緻化のための初期姿勢推定を提供することができる。それは、収束およびロバストネスを向上させることができるように、姿勢精緻化のための制約項として用いられてもよい。

20

【0080】

すると、コスト関数は以下のように定義されてよい。

$$E = E_c + \eta * E_{match} = F_1 + F_2 + \eta * \sum_{j=1}^N w_i * \|\pi(P_i) - p_i\| \quad (20)$$

ここで、は、 E_c と E_{match} との間のエネルギーのバランスをとる重み関数である。

我々は E_c および E_{match} の両方についての勾配流を導出することになるが、最初の実行では、 $\eta = 0$ として、 E_c のエネルギーを用いるだけである。

30

【0081】

姿勢を反復的に更新するために速度スクリュウを計算する勾配降下法に従い、エネルギー E は最小値に近づくことが期待される。

姿勢パラメーターまたは、勾配降下を用いて行われる E を最小化する速度スクリュウ $\dot{r} = [V; \Omega]^T = [v_x, v_y, v_z, w_x, w_y, w_z]^T$ を反復的に更新する。

【0082】

姿勢パラメーター r に関する E の偏微分は、連鎖法則を用いて算出され得る。

$$E_c = \int T(x, y) * (I(x, y) - u_1)^2 dx dy + \int (1 - T(x, y)) * (I(x, y) - u_2)^2 dx dy$$

$$E_{match} = \sum_{j=1}^N w_j * \|\pi(P_j) - p_j\| = \sum_{j=1}^M w_i * ((u_j - p_{jx})^2 + (v_j - p_{jy})^2)$$

40

【0083】

E_c の偏微分

まず E_c の偏微分を導出する

$$\begin{aligned}
\frac{\partial E_c}{\partial r_i} &= \int \frac{\partial T(x,y)}{\partial r_i} (I(x,y)-u_1)^2 dx dy + \int T(x,y) * 2 * (I(x,y)-u_1) \frac{\partial u_1}{\partial r_i} dx dy \\
&- \int \frac{\partial T(x,y)}{\partial r_i} (I(x,y)-u_2)^2 dx dy + \int (1-T(x,y)) * 2 * (I(x,y)-u_2) \frac{\partial u_2}{\partial r_i} dx dy \\
&= \int \left((I(x,y)-u_1)^2 - (I(x,y)-u_2)^2 \right) \frac{\partial T(x,y)}{\partial r_i} dx dy + \int T(x,y) * 2 * (I(x,y)-u_1) \frac{\partial u_1}{\partial r_i} dx dy \\
&+ \int (1-T(x,y)) * 2 * (I(x,y)-u_2) \frac{\partial u_2}{\partial r_i} dx dy
\end{aligned} \tag{21}$$

10

第2および第3項については、それらがゼロであることを証明することができる。

$$\begin{aligned}
\int T(x,y) * 2 * (I(x,y)-u_1) \frac{\partial u_1}{\partial r_i} dx dy &= 2 * \frac{\partial u_1}{\partial r_i} \left(\int T(x,y) (I(x,y)-u_1) dx dy \right) \\
&= 2 * \frac{\partial u_1}{\partial r_i} (u_1 * A_{in} - u_1 A_{in}) = 0
\end{aligned} \tag{22}$$

ここで、 $A_{in} = \int T(x,y) dx dy$ は物体の内部の面積である。

$$\begin{aligned}
\int (1-T(x,y)) * 2 * (I(x,y)-u_2) \frac{\partial u_2}{\partial r_i} dx dy &= 2 * \frac{\partial u_2}{\partial r_i} \left(\int (1-T(x,y)) (I(x,y)-u_2) dx dy \right) \\
&= 2 * \frac{\partial u_2}{\partial r_i} (u_2 * A_{out} - u_2 A_{out}) = 0
\end{aligned} \tag{23}$$

20

ここで、 $A_{out} = \int (1-T(x,y)) dx dy$ は物体の外部の面積である。

従って、次式を得ることができる。

$$\frac{\partial E_c}{\partial r_i} = \int \left((I(x,y)-u_1)^2 - (I(x,y)-u_2)^2 \right) \frac{\partial T(x,y)}{\partial r_i} dx dy \tag{24}$$

30

2値マスク $T(x, y)$ の画像は所与の3D姿勢およびカメラパラメーターについてのモデルの投影である。

$$\frac{\partial T(x,y)}{\partial r_i} = \frac{\partial T(x,y)}{\partial x} \frac{\partial x}{\partial r_i} + \frac{\partial T(x,y)}{\partial y} \frac{\partial y}{\partial r_i} \tag{25}$$

式(12)における画像ヤコビ行列より、 $\frac{\partial x}{\partial r_i}$ および $\frac{\partial y}{\partial r_i}$ は以下のように得られる。

$$\begin{bmatrix} \frac{\partial x}{\partial r_1} & \frac{\partial x}{\partial r_2} & \frac{\partial x}{\partial r_3} & \frac{\partial x}{\partial r_4} & \frac{\partial x}{\partial r_5} & \frac{\partial x}{\partial r_6} \\ \frac{\partial y}{\partial r_1} & \frac{\partial y}{\partial r_2} & \frac{\partial y}{\partial r_3} & \frac{\partial y}{\partial r_4} & \frac{\partial y}{\partial r_5} & \frac{\partial y}{\partial r_6} \end{bmatrix} = J(r) = \begin{bmatrix} \frac{\lambda_x}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda_y} & -\frac{\lambda_x^2 + u^2}{\lambda_x} & \frac{\lambda_x v}{\lambda_y} \\ 0 & -\frac{\lambda_y}{z} & \frac{v}{z} & \frac{\lambda_y^2 + v^2}{\lambda_y} & -\frac{uv}{\lambda_y} & -\frac{\lambda_y u}{\lambda_x} \end{bmatrix} \tag{26}$$

40

$\frac{\partial T(x,y)}{\partial x}$ および $\frac{\partial T(x,y)}{\partial y}$ は $T(x, y)$ の、 x および y 方向における微分である。

【0084】

$T(x, y)$ は、式(17)において定義される通りの、モデルの投影の2値マスクであるので。境界 C 上を除いて $T(x, y)$ の勾配はゼロである。

50

$$\nabla T(x, y) = \begin{bmatrix} \frac{\partial T(x, y)}{\partial x} \\ \frac{\partial T(x, y)}{\partial y} \end{bmatrix} = \begin{cases} \neq 0 & \text{if } (x, y) \in C \\ = 0 & \text{else} \end{cases}$$

$$\frac{\partial E_c}{\partial r_i} = \iint \left((I(x, y) - u_1)^2 - (I(x, y) - u_2)^2 \right) \frac{\partial T(x, y)}{\partial r_i} dx dy$$

$$= \int_{(x, y) \in C} \left((I(x, y) - u_1)^2 - (I(x, y) - u_2)^2 \right) \left(\frac{\partial T(x, y)}{\partial x} \frac{\partial x}{\partial r_i} + \frac{\partial T(x, y)}{\partial y} \frac{\partial y}{\partial r_i} \right) dx dy \quad (27)$$

10

【0085】

E_{match}の偏微分

一致エネルギーは、画像座標における、入力画像内の特徴点と投影仮想画像内の、特徴点との間の距離一致した特徴点を測るコスト関数である。

$$E_{match} = \sum_1^N w_i * \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - p_i \right\| \quad (28)$$

ここで

$$u = \lambda_x \frac{X_c}{Z_c} \quad \text{および} \quad v = \lambda_x \frac{Y_c}{Z_c} \quad \text{であり、}$$

20

カメラフレームに対する（固定点）Pの速度は次式によって与えられる。

$$\dot{P} = -\Omega \times P - V$$

【0086】

画像ヤコビ行列の微分ステップと同様に、次式を得ることができる。

$$\dot{u} = -\frac{\lambda_x}{Z_c} v_x + \frac{u}{Z_c} v_z + \frac{uv}{\lambda_y} w_x - \frac{\lambda_x^2 + u^2}{\lambda_x} w_y + \frac{\lambda_x v}{\lambda_y} w_z$$

$$\dot{v} = -\frac{\lambda_y}{Z_c} v_y + \frac{v}{Z_c} v_z + \frac{\lambda_y^2 + v^2}{\lambda_y} w_x - \frac{uv}{\lambda_y} w_y - \frac{\lambda_y u}{\lambda_x} w_z$$

30

$$\frac{\partial E_{match}}{\partial r_i} = \sum_{j=1}^N w_j * 2 \left((u_j - p_{jx}) \frac{\partial u_j}{\partial r_i} + (v_j - p_{jy}) \frac{\partial v_j}{\partial r_i} \right) \quad (29)$$

$\frac{\partial u_j}{\partial r_i}$ および $\frac{\partial v_j}{\partial r_i}$ は以下のように画像ヤコビ行列の形式で書き直され得る。

$$\begin{bmatrix} \frac{\partial u}{\partial r_1} & \frac{\partial u}{\partial r_2} & \frac{\partial u}{\partial r_3} & \frac{\partial u}{\partial r_4} & \frac{\partial u}{\partial r_5} & \frac{\partial u}{\partial r_6} \\ \frac{\partial v}{\partial r_1} & \frac{\partial v}{\partial r_2} & \frac{\partial v}{\partial r_3} & \frac{\partial v}{\partial r_4} & \frac{\partial v}{\partial r_5} & \frac{\partial v}{\partial r_6} \end{bmatrix} = J(r) = \begin{bmatrix} -\frac{\lambda_x}{Z_c} & 0 & \frac{u}{Z_c} & \frac{uv}{\lambda_y} & -\frac{\lambda_x^2 + u^2}{\lambda_x} & \frac{\lambda_x v}{\lambda_y} \\ 0 & -\frac{\lambda_y}{Z_c} & \frac{v}{Z_c} & \frac{\lambda_y^2 + v^2}{\lambda_y} & -\frac{uv}{\lambda_y} & -\frac{\lambda_y u}{\lambda_x} \end{bmatrix} \quad (30)$$

40

【0087】

部分方程式は画像およびカメラ座標で導出されており、3D特徴点 $P_j = (X_j^{obj}, Y_j^{obj}, Z_j^{obj})$ は、部分方程式を計算する際にはカメラ座標に変換されることになることに留意されたい。

50

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R_{co} \begin{pmatrix} X_{obj} \\ Y_{obj} \\ Z_{obj} \end{pmatrix} + T_{co} \quad (31)$$

【0088】

エネルギー関数を最小化するためのガウス・ニュートン法
我々の実装では、エネルギー関数を最小化するためにガウス・ニュートン法を用いた。
ガウス・ニュートンアルゴリズムは2乗和の最小値を求める。

10

$$E = \sum_1^M e^2(r) \quad (32)$$

最小値のための最初の推測より開始し、反復によって方法は進行する。

$$r^n = r^{n-1} + \Delta r \quad (33)$$

ここで、増分 Δr は正規方程式に対する解である。

$$e = -J_r \Delta r \quad (34)$$

2乗和を最小化する Δr を求める作業、

$$\min \|e + J_r \Delta r\|$$

20

すると、増分 Δr は次式によって計算される。

$$\Delta r = -(J_r^T J_r)^{-1} J_r^T e \quad (35)$$

【0089】

エネルギー E_c を最小化する

各輪郭点 $(x, y) \in C$ について、以下のようなになる。

$$e_c = (I(x, y) - u_1)^2 - (I(x, y) - u_2)^2$$

および

$$J_c = [J_1, J_2, J_3, J_4, J_5, J_6]$$

30

および

$$i=1\dots6 \text{ について、 } J_i = \frac{\partial T(x, y)}{\partial x} \frac{\partial x}{\partial r_i} + \frac{\partial T(x, y)}{\partial y} \frac{\partial y}{\partial r_i}$$

【0090】

エネルギー E_{match} を最小化する

一致した各特徴点 j について、

$$e_{match} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} u_j - p_{jx} \\ v_j - p_{jy} \end{bmatrix} * w_j$$

および

$$J_{match} = \begin{bmatrix} -\frac{\lambda_x}{Z_c} & 0 & \frac{u}{Z_c} & \frac{uv}{\lambda_y} & -\frac{\lambda_x^2 + u^2}{\lambda_x} & \frac{\lambda_x v}{\lambda_y} \\ 0 & -\frac{\lambda_y}{Z_c} & \frac{v}{Z_c} & \frac{\lambda_y^2 + v^2}{\lambda_y} & -\frac{uv}{\lambda_y} & -\frac{\lambda_y u}{\lambda_x} \end{bmatrix} * w_j \quad 10$$

【 0 0 9 1 】

エネルギー $E = E_c + E_{match}$ を最小化する

e および J は、 e_c 、 e_{match} ならびに J_c および J_{match} をそれぞれ連結することによって得られる。

$$e = \begin{bmatrix} e_c \\ \eta * e_{match} \end{bmatrix} \quad 20$$

および

$$J = \begin{bmatrix} J_c \\ \eta * J_{match} \end{bmatrix}$$

増分 Δr は次式によって計算されることになる。

$$\Delta r = -(J^T J)^{-1} J^T e$$

【 0 0 9 2 】

姿勢パラメータまたは変換行列を更新する

30

物体座標からカメラ座標への変換は、変換による表示であってよい。

$$X(t) = R(t)X_0 + T(t) \quad (37)$$

斉次の表示では、変換行列は、

$$g(t) = \begin{bmatrix} R(t) & T(t) \\ 0 & 1 \end{bmatrix} \quad \text{および} \quad X(t) = g(t)X_0 \quad (38)$$

時刻 t における或る固定フレーム（物体座標）とカメラフレームとの間の相対変位を示すために $g(t) = (R(t), T(t))$ を用いることにする。カメラが時刻 t_1, t_2, \dots, t_m においてそれぞれロケーション $g(t_1), g(t_2), \dots, g(t_m)$ にあるならば、このとき、同じ点 p の座標はそれに応じて $X(t_i) = g(t_i)X_0, i = 1, \dots, m$ として与えられる。

【 0 0 9 3 】

図 7 を参照すると、 t_2 および t_1 におけるカメラの間の相対運動は $g(t_2, t_1)$ として示されることとなり、このとき、同じ点 p の座標同士の間には以下の関係がある。

$$X(t_2) = g(t_2, t_1)X(t_1) \quad (39)$$

【 0 0 9 4 】

今度は、図 7 に示される通りの、 $t = t_3$ におけるカメラの第 3 の位置を考える。 t_3 と t_2 とにおけるカメラの間の相対運動は $g(t_3, t_2)$ であり、 t_3 と t_1 とにおけるカメ

50

ラの間での相対運動は $g(t_3, t_1)$ である。このとき、座標の間には以下の関係がある。

$$X(t_3) = g(t_3, t_2) X(t_2) = g(t_3, t_2) g(t_2, t_1) X(t_1) \quad (40)$$

【0095】

t_3 および t_1 における座標の間の直接的関係と比較すると

$$X(t_3) = g(t_3, t_1) X(t_1) \quad (41)$$

【0096】

連続運動のための合成規則が成り立たなければならない。

$$g(t_3, t_1) = g(t_3, t_2) g(t_2, t_1) \quad (42)$$

【0097】

合成規則に基づけば、 $g(t_n, t_1)$ についての姿勢パラメータは

$$g(t_n, t_1) = g(t_n, t_{n-1}) g(t_{n-1}, t_1) \quad (43)$$

【0098】

勾配降下法より、反復 $n - 1$ から n への相対変位を得ることができる。その結果、反復 n のための姿勢パラメータ更新は、 $g(t_n, t_{n-1})$ を計算することによって得られる。

$$g(t_n, t_{n-1}) = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \quad (44)$$

【0099】

相対変位は反復における制御入力（速度スクリュー）

$\dot{r} = [V; \Omega]^T = [v_x, v_y, v_z, w_x, w_y, w_z]^T$ から得られる。収束速度を調整するためにステップサイズ (stepsize) 関数が用いられる。

$$\Delta r = [v_x, v_y, v_z, w_x, w_y, w_z]^T * \text{stepsize} = [T_x, T_y, T_z, \phi, \theta, \psi] \quad (45)$$

$$R = R_z * R_y * R_x$$

および

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

および

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

【0100】

10

20

30

40

従って、次式を得ることができ、

$$g(t_n, t_{n-1}) = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_z R_y R_x & T \\ 0 & 1 \end{bmatrix} \quad (46)$$

さらに、物体座標からカメラ座標への新しい変換行列は次式のようになる。

$$g(t_n, t_1) = g(t_n, t_{n-1})g(t_{n-1}, t_1) \quad (47)$$

【0101】

実行

10

姿勢推定はエネルギー関数の最適化または最小化としてモデル化される。反復的に姿勢更新は計算され、エネルギーを最小化することになる。以下は、上述された方法の例示的な実行である。

ステップ1：画像、モデル、カメラパラメータを入力する。

ステップ2：初期姿勢パラメータを得る。

ステップ3：与えられた姿勢パラメータを用いてOpenGLによってモデルを仮想画像に投影する。2値マスク画像 $T(x, y)$ および対応する深さ情報/マップが得られる。

ステップ4：姿勢を反復的に更新するために速度スクリュウを計算する。(i)式(18)によって物体の内部および外部の平均値を計算する。(ii)2値マスクから輪郭Cを得、深さ情報を用いて3D輪郭点を得る。(iii)輪郭点についての、xおよびy方向における勾配 $T(x, y)$ を計算する。(iv)式(26)を用いて画像ヤコビ行列を計算する。(v)式(27)を用いて勾配流を計算し、ガウス・ニュートン法を用いて式(35)によって制御入力(速度スクリュウ)を得る。

20

ステップ5：式(45)によって、制御入力(速度スクリュウ)および所与のステップサイズを用いて相対変位を計算する。

ステップ6：式(46)および(47)によって姿勢パラメータを更新する。

ステップ7：もし、姿勢パラメータの変化が非常に小さいか、または最大反復数に達すれば、終了に行き、さもなければステップ3に戻る。

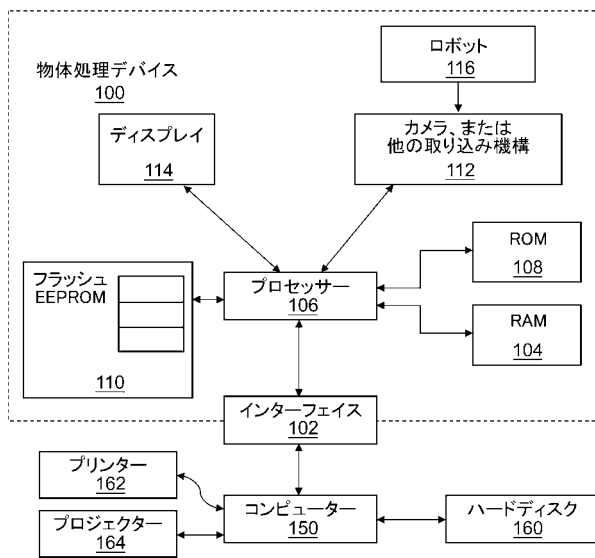
ステップ8：終了。

30

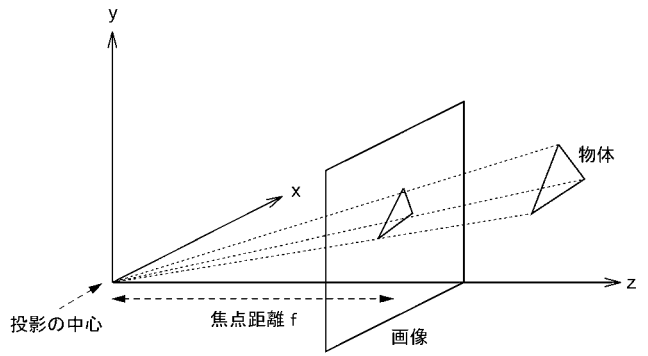
【0102】

本発明はいくつかの特定の実施形態と併せて記載されているが、上述の記載に鑑みれば、多くのさらなる代替、変更および変形が容易に分かることは当業者には明白である。それ故、本願明細書において記載されている本発明はこのような代替、変更、応用および変形をすべて、添付の請求項の精神および範囲内にあってよいものとして含むように意図されている。

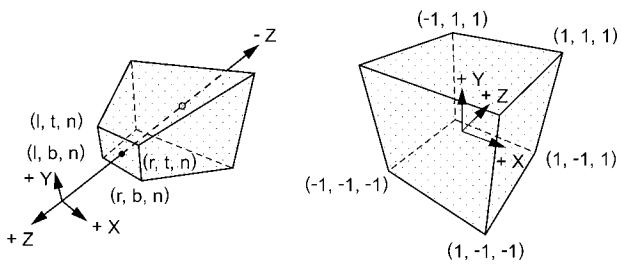
【 図 1 】



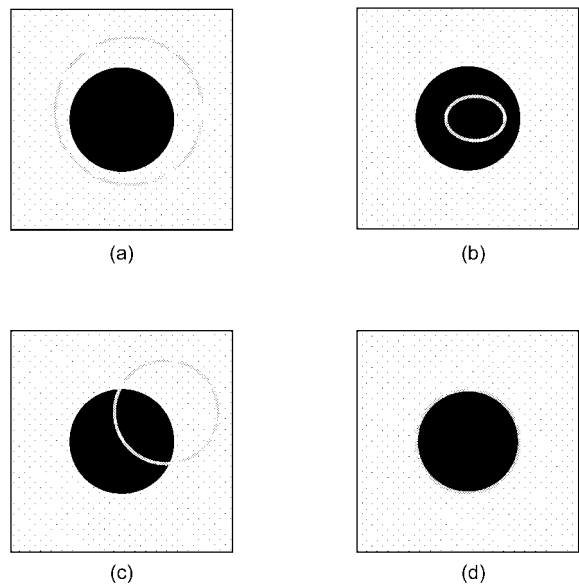
【 図 4 】



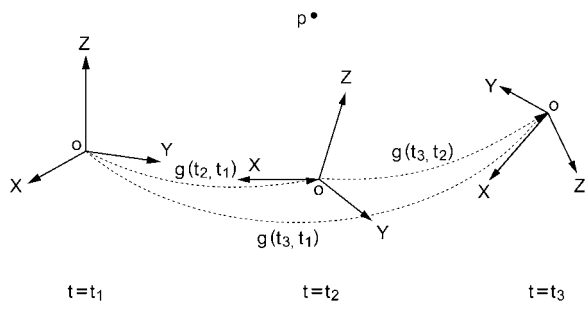
【 図 5 】



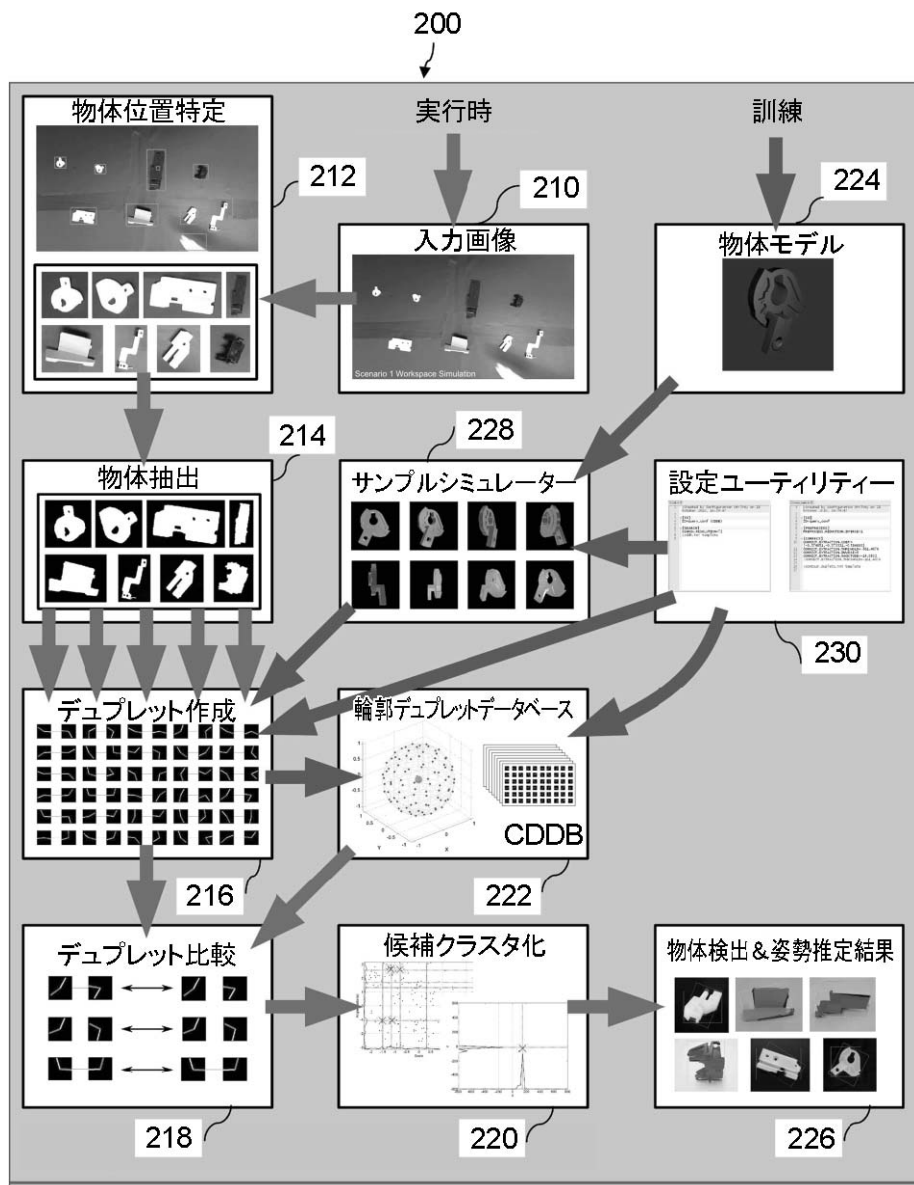
【 図 6 】



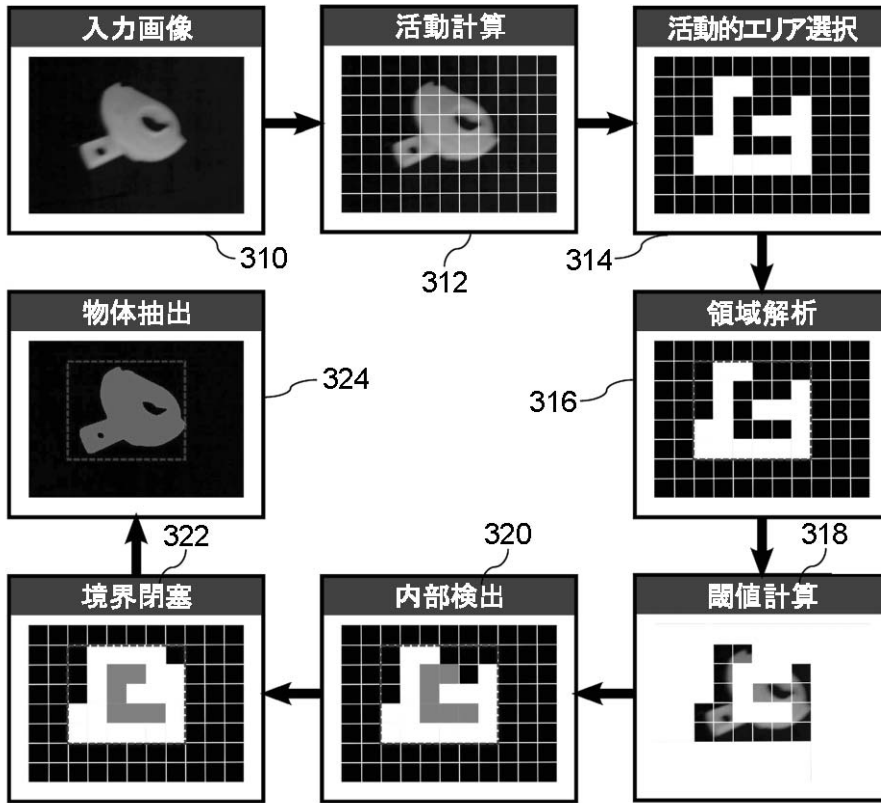
【 図 7 】



【 図 2 】



【 図 3 】



フロントページの続き

(72)発明者 グオイー フ

カナダ国 オンタリオ州 ボーン コルトレイン ドライブ 180

(72)発明者 イヴォ モラヴェク

カナダ国 オンタリオ州 スカーボロー ビクトリア パーク アベニュー 2727-708

Fターム(参考) 3C707 KS03 KS04 KT01 KT11

5B057 AA05 DA07 DC30