(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2013-137709
(P2013-137709A)
(43)公開日　平成25年7月11日(2013.7.11)

(51)Int.Cl.			F I			テーマコード（参考）
G06F	3/041	(2006.01)	G06F	3/041	380B	5B068
G09G	5/00	(2006.01)	G09G	5/00	X	5C082
G09G	5/36	(2006.01)	G09G	5/36	530Y	
G09G	5/377	(2006.01)	G09G	5/00	530M	
G09G	5/38	(2006.01)	G09G	5/36	520L	

審査請求 未請求　請求項の数 7　ＯＬ　（全 14 頁）　最終頁に続く

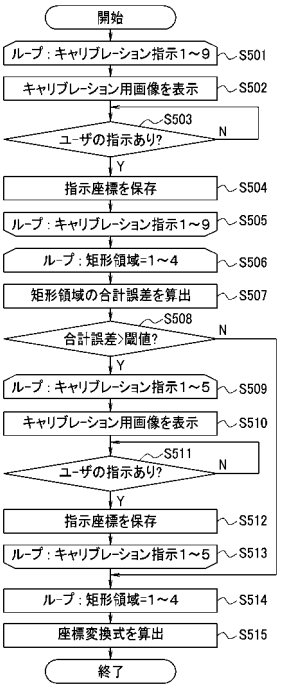
(21)出願番号　特願2011-289082 (P2011-289082)
(22)出願日　平成23年12月28日 (2011.12.28)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100090273
弁理士　國分　孝悦
(72)発明者　松下　明弘
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

Ｆターム(参考)　5B068 AA01 BB18 BE04 DD11
5C082 AA03 AA14 AA24 AA27 AA34 BA12 BA13 BA27 BB01 BB42 BD02 BD07 CA02 CA33 CA34 CA40 CA42 CA54 CA56 CA76 CA82 CA85 CB01 CB03 CB06 CB08 CB10 DA42 DA87 MM05 MM09

(54)【発明の名称】キャリブレーション装置、その制御方法及びプログラム

(57)【要約】
【課題】ユーザの作業負荷を抑えつつ、効率よく誤差を低減させる。
【解決手段】ＣＰＵは、キャリブレーションの際にユーザに指示させるポイントを含む画像データを画面上に表示させる（Ｓ５０２）。次にＣＰＵは、ユーザにより指示された画面上における位置を検出する（Ｓ５０３）。次にＣＰＵは、画像データ上におけるポイントの位置と、ユーザにより指示された画面上における位置との誤差を算出し（Ｓ５０７）、当該誤差に応じて、画像データに含まれるポイントを追加する（Ｓ５１０）。
【選択図】図５



【特許請求の範囲】
【請求項１】
キャリブレーションの際にユーザに指示させるポイントを含む画像データを画面上に表示させる表示制御手段と、
ユーザにより指示された前記画面上における位置を検出する検出手段と、
前記画像データ上における前記ポイントの位置と、ユーザにより指示された前記画面上における位置との誤差を算出する誤差算出手段と、
前記誤差算出手段により算出された誤差に応じて、前記画像データに含まれるポイントを追加する追加手段とを有することを特徴とするキャリブレーション装置。
【請求項２】
前記誤差算出手段は、前記画像データの部分領域毎に、前記画像データ上における前記ポイントの位置と、ユーザにより指示された前記画面上における位置との誤差を算出し、前記追加手段は、前記誤差算出手段により部分領域毎に算出された誤差に応じて、前記画像データの各部分領域におけるポイントを追加することを特徴とする請求項１に記載のキャリブレーション装置。
【請求項３】
前記ポイントは複数のポイントであり、前記部分領域は、前記複数のポイントによって分割された領域であることを特徴とする請求項２に記載のキャリブレーション装置。
【請求項４】
前記検出手段により検出された、ユーザにより指示された前記画面上における一部の位置に基づいて、前記画像データ上における前記ポイントの位置を算出する位置算出手段を更に有することを特徴とする請求項１乃至３の何れか１項に記載のキャリブレーション装置。
【請求項５】
前記画像データ上における前記ポイントの位置は、予め定められた位置であることを特徴とする請求項１乃至３の何れか１項に記載のキャリブレーション装置。
【請求項６】
キャリブレーションの際にユーザに指示させるポイントを含む画像データを画面上に表示させる表示制御ステップと、
ユーザにより指示された前記画面上における位置を検出する検出ステップと、
前記画像データ上における前記ポイントの位置と、ユーザにより指示された前記画面上における位置との誤差を算出する誤差算出ステップと、
前記誤差算出ステップにより算出された誤差に応じて、前記画像データに含まれるポイントを追加する追加ステップとを有することを特徴とするキャリブレーション装置の制御方法。
【請求項７】
キャリブレーションの際にユーザに指示させるポイントを含む画像データを画面上に表示させる表示制御ステップと、
ユーザにより指示された前記画面上における位置を検出する検出ステップと、
前記画像データ上における前記ポイントの位置と、ユーザにより指示された前記画面上における位置との誤差を算出する誤差算出ステップと、
前記誤差算出ステップにより算出された誤差に応じて、前記画像データに含まれるポイントを追加する追加ステップと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、キャリブレーションの際にユーザに指示させるポイントを表示させる技術に関するものである。
【背景技術】
【０００２】

会議やプレゼンテーション等では、多人数で情報を共有するために、コンピュータの画像データをプロジェクタに入力して、スクリーンに投影表示させることが一般的に行われている。このとき、ユーザが例えばペンや指等でスクリーンに表示された画像上を指示することにより、接続されたコンピュータの操作を行うことが可能な座標入力システムが広く利用されるようになってきている。座標入力システムとしては、光を用いた方式、超音波を用いた方式及び抵抗膜を用いた方式等の各種方式がある。

【０００３】

ここで、スクリーンに座標入力を検出する装置を取りつけて、プロジェクタから画像を投影して利用する場合を考える。このとき、一般的に、ユーザによって指示された座標入力システム上の座標と表示される画像上の座標とを一致させるために、キャリブレーションと呼ばれる作業を行う必要がある。キャリブレーションは、コンピュータにインストールされたアプリケーション等によって実行される。表示される画像上の特定の座標に例えば十字等の形状をした画像を表示し、その交叉点をユーザに指示してもらうことにより、座標入力システム上の座標と表示される画像上の座標との対応付けが行われる。この作業を複数の位置で行うことにより、表示される画像全体に対して、座標入力システム上の座標との対応付けを行うことができる。

【０００４】

例えば、特許文献１においては、画面上に５つの基準点を表示して、キャリブレーションを行う方法が開示されている。しかし、実際には、正確にキャリブレーションが行われたとしても、座標入力システムの座標検出の性能や投影された画像の歪み等のために、座標入力システム上の座標と表示される画像上の座標とが場所によってずれてしまう場合がある。

【０００５】

キャリブレーションを行うポイントの数を増やして座標の対応付けを細かくすれば、ずれ量を低減することができる。しかし一方で、ユーザの作業が増えて煩わしくなるというデメリットもあるため、キャリブレーションを行うポイントの数は必要最小限にしたいという要求がある。従来、キャリブレーションを行うポイントの数は、それぞれの座標入力システムの性能等に基づいて決められていた。また、キャリブレーションを行うポイントの数をユーザが選択できるようにした座標入力システムも開発されている。

【先行技術文献】

【特許文献】

【０００６】

【特許文献１】特開２００８－２７６３１０号公報

【発明の概要】

【発明が解決しようとする課題】

【０００７】

しかしながら、キャリブレーション作業を正確に行ったとしても、上述したように、座標入力システムの座標検出の性能や投影された画像データの歪み等のために、場所によって、座標入力システム上の座標と表示される画像上の座標との間に誤差が発生する場合がある。

【０００８】

どの程度まで誤差を抑えることができるかは、基本的にはキャリブレーションを行うポイントの数に応じて決まってくる。そのため、ポイントの数が一定であると、誤差が生じている部分があっても、誤差を低減できるレベルには一定の限界がある。また、キャリブレーションを行うポイントの数をユーザが選択するという方法もある。しかし、この方法では、一旦キャリブレーションを行った後、ユーザが使用した上で誤差が大きいと判断すると、ポイントの数を増やしてキャリブレーションの作業をもう一度やり直すことになり、煩わしい。また、画面全体に渡って多数のポイントを指示するとなると、その作業はユーザにとって大きな負担となる。

【０００９】

そこで、本発明の目的は、ユーザの作業負荷を抑えつつ、効率よく誤差を低減させることにある。

【課題を解決するための手段】

【００１０】

本発明のキャリブレーション装置は、キャリブレーションの際にユーザに指示させるポイントを含む画像データを画面上に表示させる表示制御手段と、ユーザにより指示された前記画面上における位置を検出する検出手段と、前記画像データ上における前記ポイントの位置と、ユーザにより指示された前記画面上における位置との誤差を算出する誤差算出手段と、前記誤差算出手段により算出された誤差に応じて、前記画像データに含まれるポイントを追加する追加手段とを有することを特徴とする。

【発明の効果】

【００１１】

本発明によれば、ユーザの作業負荷を抑えつつ、効率よく誤差を低減させることが可能となる。

【図面の簡単な説明】

【００１２】

【図１】本発明の実施形態に係る座標入力システムの一部構成を示す図である。

【図２】センサユニットと制御ユニットとの構成を示す図である。

【図３】ＣＣＤからＣＰＵに入力されるデータをグラフ化して示した図である。

【図４】指示があった座標入力システム上の座標の算出方法を説明するための図である。

【図５】キャリブレーションプログラムによって実行される処理を示すフローチャートである。

【図６】キャリブレーション用画像の例を示す図である。

【発明を実施するための形態】

【００１３】

以下、本発明を適用した好適な実施形態を、添付図面を参照しながら詳細に説明する。なお、以下に説明する実施形態は飽くまでも本発明の一適用例に過ぎず、本発明は以下の実施形態に限定されるものではない。

【００１４】

先ず、本発明の第１の実施形態について説明する。図１は、本発明の第１の実施形態に係る座標入力システムの一部構成を示す図である。図１において、１はホワイトボードであり、プロジェクタによって投影される画像を表示させるためのスクリーンとして使用される。外部のコンピュータがプロジェクタに接続されており、コンピュータから出力される画像データに対応する画像がホワイトボード１に対して投影されるようになっている。２は、プロジェクタから投影された画像の表示領域である。３は、指等でスクリーン上が指示されたときに、指示された座標を検出するためのデジタイザユニットであり、図１に示すようにホワイトボード１の左右及び下辺を囲むような形状となっている。デジタイザユニット３は、ホワイトボード１の表面に固定されている。

【００１５】

４及び５は、投光を行うためのＬＥＤと受光のためのセンサとを備えたセンサユニットであり、それぞれデジタイザユニット３内部の左右の上端部分に配置されている。７は、再帰反射部であり、入射光を到来方向に反射する再帰反射面を有する。再帰反射部７は、ホワイトボード１の左右及び下辺を囲むように構成されており、センサユニット４及び５から投光された光を、投光した各々のセンサユニット４及び５に向けて再帰的に反射する。反射された光は、集光光学系及びＣＣＤ等によって構成されたセンサユニット４及び５内部のセンサによって検出され、その光量分布を示すデータ（光量分布データ）が出力される。

【００１６】

また、デジタイザユニット３内部の下辺部分には制御ユニット６が配置されている。制御ユニット６は、ＣＰＵ及びメモリ等の周辺回路を備える。制御ユニット６は、センサユ

ニット４及び５のコントロールや、センサユニット４及び５からの出力データの受信及び座標計算等の処理を行う。

【００１７】

図２は、センサユニット４及び５と制御ユニット６との構成を示す図である。図２に示すように、制御ユニット６は、座標入力システムの各種制御を行うためのＣＰＵ８と、ＣＰＵ８を動作させるためのプログラムが記録されているフラッシュメモリ１０と、ＣＰＵ８のワーク用メモリであるＳＲＡＭ９とを備える。また制御ユニット６は、ＣＰＵ８によって算出された座標データを、外部に接続されたコンピュータ１８に送信するためのＵＳＢインタフェース部１１を備える。なお、センサユニット４及び５と制御ユニット６とは、キャリブレーション装置の適用例となる構成である。

【００１８】

センサユニット４及び５はそれぞれ、投光を行うためのＬＥＤ１２及び１５と、受光のためのセンサであるＣＣＤ１３及び１６とを備える。本実施形態では、ＣＣＤ１３及び１６の出力はアナログ信号である。また、センサユニット４及び５は、ＣＣＤ１３及び１６から出力されるアナログ信号をデジタル信号に変換するＡ／Ｄ変換部１４及び１７を備える。Ａ／Ｄ変換部１４及び１７から出力されるデジタル信号は、ＣＰＵ８に入力される。

【００１９】

ＣＰＵ８は、ＣＣＤ１３及び１６のシャッタタイミングやデータの出力タイミングを制御するための信号を生成し、ＣＣＤ１３及び１６に供給する。またＣＰＵ８は、ＬＥＤ１２及び１５を駆動するための信号を生成し、ＬＥＤ１２及び１５に供給する。またＣＰＵ８は、ＬＥＤ１２を点灯するための信号を出力するとともに、ＣＣＤ１３のシャッタを制御する信号を出力する。これにより、ＣＣＤ１３は、ＬＥＤ１２から投光された光が再帰反射部７で反射された光を検出する。またＣＰＵ８は、ＬＥＤ１５を点灯するための信号を出力するとともに、ＣＣＤ１６のシャッタを制御する信号を出力する。これにより、ＣＣＤ１６は、ＬＥＤ１５から投光された光が再帰反射部７で反射された光を検出する。

【００２０】

ＣＣＤ１３及び１６の出力信号は、Ａ／Ｄ変換部１４及び１７でデジタル化され、ＣＰＵ８に入力される。図３は、ＣＣＤ１３及び１６からＣＰＵ８に入力されるデータをグラフ化して示した図である。例えば、スクリーン上への指示がない場合、図３（ａ）に示すような光量分布データが得られる。横軸はＣＣＤの画素番号であり、縦軸はＣＣＤが出力する光量レベルである。得られる光量分布データの形状は、主にＬＥＤ１２及び１５の投光特性、再帰反射部７の再帰反射特性、並びに、センサユニット４及び５から再帰反射部７までの距離等によって決定される。図３（ｂ）は、指等によってスクリーン上への指示が行われた状態での光量分布データを示している。ＬＥＤ１２及び１５から投光された光が遮られ、再帰反射部７による反射光が得られなくなるため、遮られた部分の光量が低下する。

【００２１】

本座標入力システムの電源を入れた直後等に、ＣＰＵ８は、光量分布データを取得する。即ち、ＣＰＵ８は、ＬＥＤ１２及び１５からの投光を行わない状態の光量分布データＢ［ｘ］（ｘは１～Ｎ）を取得し、ＳＲＡＭ９に記憶する。なお、光量分布データＢ［ｘ］は、図３（ａ）における破線部分のデータに相当する。ここで、ｘはＣＣＤ１３及び１６の画素番号であり、ＮはＣＰＵ８が読み込むＣＣＤ１３及び１６の画素の総数である。

【００２２】

次にＣＰＵ８は、ＬＥＤ１２及び１５からの投光を行った状態で、図３（ａ）の実線部分のようなスクリーン上への指示がない状態での光量分布データをリファレンス光量分布データＲｅｆ［ｘ］として取得し、ＳＲＡＭ９に記憶する。その後、ＣＰＵ８は、一定期間毎に光量分布データＬ［ｘ］を取得して、スクリーンに対する指示の有無を検出する。指等によってスクリーン上への指示が行われると、図３（ｂ）に示すように、遮られた部分の光量が低下する。ＣＰＵ８は、リファレンス光量分布データＲｅｆ［ｘ］と光量分布データＬ［ｘ］との差分を算出することにより、この変化を検出する。即ち、ＣＰＵ８は

、次の式１を用いて、各画素番号について遮光された割合Ｋ［ｘ］を算出する。
Ｋ［ｘ］＝（Ｒｅｆ［ｘ］－Ｌ［ｘ］）／（Ｒｅｆ［ｘ］－Ｂ［ｘ］）・・・式１
ＣＰＵ８は、Ｋ［ｘ］の値が予め決められた閾値Ｔｈより大きい場合、指示があったと判定する。指示があった場合、ＣＰＵ８は、当該指示があったと判定した画素番号の範囲を検出し、その範囲の中央を指示の中心とする。次にＣＰＵ８は、テーブルを参照することにより、指示の中心と判定した画素番号を角度θに変換する。

【００２３】

図４は、指示があった座標入力システム上の座標の算出方法を説明するための図である。先ずＣＰＵ８は、左側に配置されたセンサユニット４の位置を原点として、そこから右側のセンサユニット５を通る方向をｘ軸とする。またＣＰＵ８は、左側のセンサユニット４を通り、ｘ軸に垂直な方向をｙ軸とする。ＣＰＵ８は、左側のセンサユニット４の座標を原点（０，０）とし、右側のセンサユニット５の座標を（１，０）として、座標入力システム上の座標系を定める。ここで、座標入力システム上の座標系において、座標（ｐｘ，ｐｙ）に指示があったものとする。上述した手順（テーブル参照）により指示の中心と判定された画素番号に対応する角度θ０及びθ１が求められる。ここで、角度が０°となる基準方向をｘ軸の方向とする。ｘ軸の方向は、例えば本座標入力システムの電源が入れられたときに片方のセンサユニットのＬＥＤを発光させ、その光を反対側のセンサユニットによって検出することによって決定される。なお、指示があった座標（ｐｘ，ｐｙ）は、次の式２及び式３を用いて算出される。

ｐｘ＝ｔａｎθ１／（ｔａｎθ０＋ｔａｎθ１）・・・式２
ｐｙ＝ｔａｎθ０×ｔａｎθ１／（ｔａｎθ０＋ｔａｎθ１）・・・式３

【００２４】

式２及び式３により算出された座標入力システム上の座標（ｐｘ，ｐｙ）は、ＵＳＢインタフェース部１１に入力されてＵＳＢのコマンドに変換され、座標変換等の座標処理を行うコンピュータ１８に送信される。コンピュータ１８には、予め本座標入力システムを使用するために必要となるドライバ及びアプリケーションがインストールされている。座標（ｐｘ，ｐｙ）は、コンピュータ１８に入力されると、ドライバに送られる。ドライバは、座標入力システム上の座標（ｐｘ，ｐｙ）を、コンピュータ１８の画像データ上の座標（ｈｘ，ｈｙ）に変換する。

【００２５】

ここで、座標入力システム上の座標（ｐｘ，ｐｙ）を、コンピュータ１８の画像データ上の座標（ｈｘ，ｈｙ）に変換するために必要なパラメータを得るためには、予めキャリブレーションを行っておく必要がある。キャリブレーションは、例えば本座標入力システムを設置して初めて使用するときに行われる。また、座標入力システム上の座標（ｐｘ，ｐｙ）とコンピュータ１８の画像データ上の座標（ｈｘ，ｈｙ）との対応関係は、デジタイザユニット３が設置された位置と、プロジェクタから投影された画像の表示領域２の位置や大きさとの関係によって変わる。従って、デジタイザユニット３の設置位置を変えたり、プロジェクタから投影された画像の表示領域２の位置や大きさを変えたりして対応関係が変わったときには、キャリブレーションをやり直す必要がある。

【００２６】

キャリブレーションを行うためのプログラム（以下、キャリブレーションプログラムと称す）は、アプリケーションとしてコンピュータ１８にインストールされており、ユーザが起動することで実行される。図５は、キャリブレーションプログラムによって実行される処理を示すフローチャートである。

【００２７】

ステップＳ５０１において、ＣＰＵ８は、複数のキャリブレーションポイントで指示を行うためのループを開始する。ステップＳ５０２において、ＣＰＵ８は、コンピュータ１８から入力されたキャリブレーション用画像データに対応する画像（以下、キャリブレーション用画像と称す）を、スクリーン上の予め定められた位置に表示させる。なお、ステップＳ５０２は、表示制御手段の処理例である。図６（ａ）は、キャリブレーション用画

像の表示例を示している。ここでキャリブレーション用画像の左上端の座標が（０，０）、右下端の座標が（１９１９，１１９９）であるものとする。図６（ａ）に示すキャリブレーション用画像では、座標（２０，２０）を中心として、キャリブレーションポイントである十字形状の画像が表示されている。ステップＳ５０３において、ＣＰＵ８は、ユーザによる指示があったか否かを判定する。ユーザによる指示があった場合、処理はステップＳ５０４に移行する。一方、ユーザによる指示がない場合、処理はステップＳ５０３に戻る。

【００２８】

ステップＳ５０４において、ＣＰＵ８は、ユーザにより指示された座標を、座標入力システム上の座標として保存する。ここで、座標入力システム上のｋ（ｋは１～９）番目の座標を（ｃｘ［ｋ］，ｃｙ［ｋ］）とする。以上の処理を、座標を変えながら９箇所のキャリブレーションポイントについて繰り返す。９箇所のキャリブレーションポイントのキャリブレーション用画像上の座標は、順番に次のように決められているものとする。

（２０，２０）、（９６０，２０）、（１９００，２０）、（２０，６００）、（９６０，６００）、（１９００，６００）、（２０，１１８０）、（９６０，１１８０）、（１９００，１１８０）

【００２９】

図６（ｂ）は、キャリブレーション用画像上における９箇所のキャリブレーションポイントの位置を示している。図６（ｂ）に示すように９箇所のキャリブレーションポイントを直線で結ぶことにより、破線で示したように４個の矩形領域（部分領域）が生成される。ステップＳ５０６において、ＣＰＵ８は、各矩形領域について、誤差の程度を評価するためのループを開始する。ステップＳ５０７において、ＣＰＵ８は、各矩形領域における誤差を算出する。なお、ステップＳ５０７は、誤差算出手段の処理例である。ここで、キャリブレーション用画像上における隅の４箇所のキャリブレーションポイント（ｋ＝１，３，７，９）の位置は、それらを結ぶことにより長方形となる位置に決められている。また、隅の４箇所のキャリブレーションポイント以外の中間のキャリブレーションポイント（ｋ＝２，４，５，６，８）の位置は、隅のキャリブレーションポイントのちょうど中点に位置するように決められている。従って、隅の４箇所のキャリブレーションポイント（ｋ＝１，３，７，９）で指示された座標入力システム上の座標を基準にすると、他のキャリブレーションポイントにおいて、誤差のない場合に期待される座標入力システム上の座標は、隅の４箇所のキャリブレーションポイントの中点として算出することができる。他のキャリブレーションポイントにおける座標入力システム上の座標の期待値を（ｄｘ［ｋ］，ｄｙ［ｋ］）とすると、これらの値は、隅の４箇所のキャリブレーションポイントの座標入力システム上の座標を用いて、式４～式１３のように算出することができる。なお、他のキャリブレーションポイントにおける座標入力システム上の座標の期待値を算出する処理は、位置算出手段の処理例である。

ｄｘ［２］＝（ｃｘ［１］＋ｃｘ［３］）／２・・・式４
ｄｙ［２］＝（ｃｙ［１］＋ｃｙ［３］）／２・・・式５
ｄｘ［４］＝（ｃｘ［１］＋ｃｘ［７］）／２・・・式６
ｄｙ［４］＝（ｃｙ［１］＋ｃｙ［７］）／２・・・式７
ｄｘ［６］＝（ｃｘ［３］＋ｃｘ［９］）／２・・・式８
ｄｙ［６］＝（ｃｙ［３］＋ｃｙ［９］）／２・・・式９
ｄｘ［８］＝（ｃｘ［７］＋ｃｘ［９］）／２・・・式１０
ｄｙ［８］＝（ｃｙ［７］＋ｃｙ［９］）／２・・・式１１
ｄｘ［５］＝（ｃｘ［４］＋ｃｘ［６］）／２・・・式１２
ｄｙ［５］＝（ｃｙ［４］＋ｃｙ［６］）／２・・・式１３

【００３０】

なお、隅の４箇所のキャリブレーションポイントを基準として他のキャリブレーションポイントの値を算出するため、投影されたキャリブレーション用画像の表示領域２は、センサユニット４及び５で検出できる範囲に該当すればよい。即ち、キャリブレーション用

画像の表示領域２はセンサユニット４及び５で検出できる範囲にあれば、表示領域２の位置やサイズは任意でよく、また、表示領域２を回転した場合にも対応することができる。

【００３１】

隅の４箇所のキャリブレーションポイント以外のキャリブレーションポイントについて、座標の期待値（ｄｘ［ｋ］，ｄｙ［ｋ］）と、実際に指示された座標（ｃｘ［ｋ］，ｃｙ［ｋ］）との間に差があれば、何らかの誤差が発生していることになる。プロジェクタから投影された画像の歪みや、センサユニット４及び５が測定した角度の精度等が座標の誤差の原因となる。また、デジタイザユニット３やセンサユニット４及び５の取り付け状態によって、センサユニット４及び５の角度の基準（０°の方向）がずれると、図４に示すθ０やθ１にオフセットが発生し、誤差が発生する。

【００３２】

先ずＣＰＵ８は、図６（ｂ）に示す４個の矩形領域のうち左上の矩形領域について合計誤差Ｅを算出する。それぞれのキャリブレーションポイントにおいて、座標の期待値（ｄｘ［ｋ］，ｄｙ［ｋ］）と、実際に指示された座標（ｃｘ［ｋ］，ｃｙ［ｋ］）との間の座標単位での距離を誤差とし、矩形領域を囲む４箇所のキャリブレーションポイントにおける誤差の合計を合計誤差Ｅとする。但し、ｋ＝１のキャリブレーションポイントは基準として用いられているため、誤差は０となるので、ｋ＝２，４，５の３箇所のキャリブレーションポイントの誤差の合計を算出すればよい。従って、合計誤差Ｅは次の式１４によって求められる。

$$E=sqrt((cx[2]-dx[2])^2+(cy[2]-dy[2])^2)+sqrt((cx[4]-dx[4])^2+(cy[4]-dy[4])^2)+sqrt((cx[5]-dx[5])^2+(cy[5]-dy[5])^2)\quad\cdots\text{式}14$$

【００３３】

ここで、ｓｑｒｔ（）は平方根を求める関数である。またここでは、合計誤差Ｅとして、座標の期待値と実際に指示された座標との距離の合計の値を用いたが、他の式を用いてもよい。例えば、式１４で平方根の関数を外した式、つまり距離の平方の合計を誤差Ｅとしてもよい。これにより、平方根演算を省略されるため、計算負荷を減らすことができる。

【００３４】

ステップＳ５０８において、ＣＰＵ８は、合計誤差Ｅが閾値Ｅｔｈより大きいか否かを判定する。合計誤差Ｅが閾値Ｅｔｈより大きい場合、処理はステップＳ５０９に移行する。一方、合計誤差Ｅが閾値Ｅｔｈ以下である場合、処理はステップＳ５１４に移行する。ここで、閾値Ｅｔｈは、実際の使用上において、どの程度の誤差までが許容されるか等を評価した結果により予め決められており、固定値として保持されている値である。

【００３５】

ステップＳ５０９において、ＣＰＵ８は、誤差を減らすためにキャリブレーションポイントを追加するためのループを開始する。ここでは、キャリブレーション用画像の左上の矩形領域においてキャリブレーションポイントが追加されたものとする。図６（ｃ）は、キャリブレーションポイントを５箇所追加した場合の例を示している。追加される５箇所のキャリブレーションポイント（ｋ＝１０～１４）は、左上の矩形領域を囲む４箇所のキャリブレーションポイントの中点となっており、それらのキャリブレーション用画像上の座標は以下のとおりである。

（４９０，２０）、（２０，３１０）、（４９０，３１０）、（９６０，３１０）、（４９０，６００）

【００３６】

ステップＳ５１０において、ＣＰＵ８は、キャリブレーション用画像を表示する。同時に、ＣＰＵ８は、キャリブレーションポイントを追加するメッセージを表示する等して、ユーザに通知する。ステップＳ５１１において、ＣＰＵ８は、ユーザの指示があったか否かを判定する。ユーザの指示があった場合、処理はステップＳ５１２に移行する。ステップＳ５１２において、ＣＰＵ８は、ユーザにより指示された座標を、座標入力システム上

の座標（ｃｘ［ｋ］，ｃｙ［ｋ］）として保存する。なお、１箇所目のキャリブレーションポイントではｋ＝１０となる。ステップＳ５０９～Ｓ５１３では、５箇所のキャリブレーションポイントについて、キャリブレーション用画像の表示処理、ユーザの指示待ち処理、及び、ユーザにより指示された座標を座標入力システム上の座標（ｃｘ［ｋ］，ｃｙ［ｋ］）として保存する処理を繰り返す。なお、ここではｋは１０～１４の値をとる。以上により、ＣＰＵ８は、キャリブレーションポイントを追加するループを終了する。

【００３７】

他の矩形領域に対して、ステップＳ５０６～Ｓ５１４のループ処理、即ち、合計誤差Ｅの算出処理、合計誤差Ｅと閾値Ｅｔｈとの比較処理、及び、キャリブレーションポイントの追加判定処理を同様に行う。以上により、各矩形領域における合計誤差を検査して、合計誤差が大きい矩形領域に対してのみ、キャリブレーションポイントの追加が行わることになる。

【００３８】

以上のキャリブレーション処理によって、左上の矩形領域だけでキャリブレーションポイントの追加が行われたものとする。その場合、キャリブレーションを行った全てのキャリブレーションポイントを図示すると、図６（ｄ）に示すようになる。左上の矩形領域については、キャリブレーションポイントの追加が行われたため、破線で示したように当該矩形領域が更に４個の矩形領域に分割されている。

【００３９】

ステップＳ５１５において、ＣＰＵ８は、矩形領域を囲む４箇所のキャリブレーションポイントのキャリブレーション用画像上の座標、及び、座標入力システム上の座標（ｃｘ［ｋ］，ｃｙ［ｋ］）を用いて、座標入力システム上の座標をキャリブレーション用画像上の座標に変換するための変換式を算出する。本実施形態においては、座標入力システム上の座標（ｐｘ，ｐｙ）を、キャリブレーション用画像上の座標（ｈｘ，ｈｙ）に変換する式として、次の式１５及び式１６が算出される。

ｈｘ＝ａ×ｐｘ×ｐｙ＋ｂ×ｐｘ＋ｃ×ｐｙ＋ｄ・・・式１５

ｈｙ＝ｅ×ｐｘ×ｐｙ＋ｆ×ｐｘ＋ｇ×ｐｙ＋ｈ・・・式１６

ここで、ａ、ｂ、ｃ、ｄ、ｅ、ｆ、ｇ、ｈは係数である。

【００４０】

例えば、一番左上の矩形領域においては、矩形領域を囲む４箇所のキャリブレーションポイントのキャリブレーション用画像上の座標はそれぞれ、以下のとおりである。

（２０，２０）、（４９０，２０）、（２０，３１０）、（４９０，３１０）

【００４１】

また、これらのキャリブレーション用画像上の座標に対応する、座標入力システム上の座標（ｃｘ［ｋ］，ｃｙ［ｋ］）のｋの値は、それぞれ、１，１０，１１，１２である。４箇所のキャリブレーションポイントのキャリブレーション用画像上の座標と、対応する座標入力システム上の座標（ｃｘ［ｋ］，ｃｙ［ｋ］）とを、式１５及び式１６に代入すると、次の式１７～式２４のようになる。

２０＝ａ×ｃｘ［１］×ｃｙ［１］＋ｂ×ｃｘ［１］＋ｃ×ｃｙ［１］＋ｄ・・・式１７

２０＝ｅ×ｃｘ［１］×ｃｙ［１］＋ｆ×ｃｘ［１］＋ｇ×ｃｙ［１］＋ｈ・・・式１８

４９０＝ａ×ｃｘ［１０］×ｃｙ［１０］＋ｂ×ｃｘ［１０］＋ｃ×ｃｙ［１０］＋ｄ・・・式１９

２０＝ｅ×ｃｘ［１０］×ｃｙ［１０］＋ｆ×ｃｘ［１０］＋ｇ×ｃｙ［１０］＋ｈ・・・式２０

２０＝ａ×ｃｘ［１１］×ｃｙ［１１］＋ｂ×ｃｘ［１１］＋ｃ×ｃｙ［１１］＋ｄ・・・式２１

３１０＝ｅ×ｃｘ［１１］×ｃｙ［１１］＋ｆ×ｃｘ［１１］＋ｇ×ｃｙ［１１］＋ｈ・・・式２２

４９０＝ａ×ｃｘ［１２］×ｃｙ［１２］＋ｂ×ｃｘ［１２］＋ｃ×ｃｙ［１２］＋ｄ・・・式２３

３１０＝ｅ×ｃｘ［１２］×ｃｙ［１２］＋ｆ×ｃｘ［１２］＋ｇ×ｃｙ［１２］＋ｈ・・・式２４

【００４２】

ＣＰＵ８は、上記８個の連立方程式を解くことにより、係数ａ、ｂ、ｃ、ｄ、ｅ、ｆ、ｇ、ｈを算出し、保存する。同様に、ＣＰＵ８は、他の全ての矩形領域についてそれぞれの係数を算出して保存する。以上でキャリブレーションプログラムが終了する。

【００４３】

キャリブレーションプログラム終了後、ＣＰＵ８は、ユーザの指示を検出するための光量分布データを取得する処理を開始する。座標入力システム上の座標（ｐｘ，ｐｙ）に指示があったとすると、その座標（ｐｘ，ｐｙ）はコンピュータ１８に送信され、ドライバに送られる。

【００４４】

ドライバにおいては、先ず指示された座標（ｐｘ，ｐｙ）が図６（ｄ）の破線で分割された各矩形領域のどこに含まれるかを判定する。例えばドライバは、各矩形領域を囲む直線の式を算出し、座標（ｐｘ，ｐｙ）が各直線の内側にあるか否かを判定することにより、含まれる矩形領域を判定する。次にドライバは、座標（ｐｘ，ｐｙ）が含まれる矩形領域に対応する係数ａ、ｂ、ｃ、ｄ、ｅ、ｆ、ｇ、ｈを用いて、式１５及び式１６により、指示された座標（ｐｘ，ｐｙ）をキャリブレーション用画像上の座標（ｈｘ，ｈｙ）に変換し、オペレーティングシステムに送信する。そして、マウスカーソルの操作や、アプリケーションの操作等が行われることになる。

【００４５】

以上のように、本実施形態においては、ドライバにおいて、指示された座標入力システム上の座標（ｐｘ，ｐｙ）から、キャリブレーション用画像上の座標（ｈｘ，ｈｙ）への変換が行われる。他の実施形態として、ＣＰＵ８においてこの変換処理を行って、ＣＰＵ８からコンピュータ１８に対して、キャリブレーション用画像上の座標（ｈｘ，ｈｙ）を送信するようにしてもよい。この場合、キャリブレーションを行ったときに、ドライバからＣＰＵ８に対して、キャリブレーションポイントのキャリブレーション用画像上の座標が送信される。そしてＣＰＵ８は、式１５及び式１６の各係数を算出し、保存しておけばよい。

【００４６】

本実施形態においては、キャリブレーションポイントを追加した矩形領域については、より小さく分割された矩形領域で変換が行われるため、誤差を低減することができる。なお、本実施形態では、初めに４個の矩形領域を生成し、各矩形領域で合計誤差を検査して、次にさらに４個の矩形領域に分割するようにしたが、もちろん他にも様々なやり方がある。例えば、９個の矩形領域や１６個の矩形領域に分割する方法でもよい。座標入力システムの誤差の出方やユーザの操作の煩雑さ等を考慮して、適切な分割数を決めればよい。また、キャリブレーションポイントの追加は１段階としたが、段階を増やしてもよい。例えば、ステップＳ５１３のループが終了した時点、即ちキャリブレーションポイントを５箇所追加した後に、キャリブレーションポイントの追加によって分割された４個の矩形領域について、さらに合計誤差の検査を行う。そして、ＣＰＵ８は、合計誤差が大きな矩形領域に対して、さらにキャリブレーションポイントの追加を行う。このような処理によって矩形領域がより小さくなるため、合計誤差をより低減することができる。

【００４７】

次に、本発明の第２の実施形態について説明する。なお、第２の実施形態に係る座標入力システムの構成は、図１及び図２に示した構成と同様であるため、以下の説明においても、図１及び図２に示した符号を用いるものとする。

【００４８】

第１の実施形態では、キャリブレーションを行ったキャリブレーションポイントのうち

隅の４箇所のキャリブレーションポイントを基準として、他のキャリブレーションポイントにおける座標の期待値を算出した。そのため、投影される画像の表示領域２は、任意の位置やサイズ、回転状態等に対応することができる。これに対し、第２の実施形態に係る画像入力システムは、投影される画像の表示領域２が固定的な画像入力システムである。例えば、１が液晶ディスプレイになっていて、投影される画像の表示領域２が固定されているような場合である。この場合、キャリブレーションを行った各キャリブレーションポイントにおける座標の期待値（ｄｘ［ｋ］，ｄｙ［ｋ］）は、予め決められた値として持っていればよいので、式４～式１３の計算を行う必要がなくなる。但し、合計誤差Ｅの計算においては、矩形領域を囲む４箇所のキャリブレーションポイントにおける誤差を合計する必要があるので、式１４の代わりに、次の式２５を用いる。

Ｅ＝ｓｑｒｔ（（ｃｘ［１］－ｄｘ［１］）^{2}＋（ｃｙ［１］－ｄｙ［１］）^{2}）＋ｓｑｒｔ（（ｃｘ［２］－ｄｘ［２］）^{2}＋（ｃｙ［２］－ｄｙ［２］）^{2}）＋ｓｑｒｔ（（ｃｘ［４］－ｄｘ［４］）^{2}＋（ｃｙ［４］－ｄｙ［４］）^{2}）＋ｓｑｒｔ（（ｃｘ［５］－ｄｘ［５］）^{2}＋（ｃｙ［５］－ｄｙ［５］）^{2}）・・・式２５

【００４９】

第２の実施形態に係る座標入力システムでは、表示領域２が固定されているため、キャリブレーション用画像の歪みは発生しない。誤差の要因としては、センサユニット４及び５が測定した角度の誤差や、デジタイザユニット３やセンサユニット４及び５の取り付け状態による、センサユニット４及び５の角度の基準（０°の方向）のずれ等が考えられる。

【００５０】

次に、本発明の第３の実施形態について説明する。上述した第１及び第２の実施形態では、デジタイザユニット３と、接続されたコンピュータ１８と、コンピュータ１８にインストールされたドライバ及びアプリケーションから構成されるシステムによって必要な機能を実現した。これに対し、第３の実施形態では、デジタイザユニット及びプロジェクタで構成される座標入力システムについて説明する。

【００５１】

第３の実施形態におけるデジタイザユニットは、ＵＳＢインタフェースを介して、表示領域に画像を投影するプロジェクタに接続されている。先ず、デジタイザユニットにおいて、ユーザによって指示された座標入力システム上の座標を算出方法は、第１及び第２の実施形態と同様である。一方、プロジェクタにはシステムコントローラが内蔵されており、プロジェクタの各種制御を行うためのプログラムが実行される。また、プロジェクタは、外部のコンピュータと接続されているものとする。キャリブレーションを行うためのプログラムは、プロジェクタ内部のシステムコントローラによって実行される。キャリブレーションプログラムの処理は図５に示したとおりである。このとき、キャリブレーション用画像は、システムコントローラが生成して、プロジェクタから投影されて表示される。そして、システムコントローラは、キャリブレーション作業が完了したとき、式１５及び式１６の係数を算出する。

【００５２】

デジタイザユニットがユーザによる指示を検出すると、座標入力システム上の座標（ｐｘ，ｐｙ）の値は、ＵＳＢインタフェース部を介してＵＳＢのコマンドに変換され、プロジェクタに送信される。プロジェクタ内部のシステムコントローラは、座標（ｐｘ，ｐｙ）を受け取ると、どの矩形領域に含まれるかを判定し、式１５及び式１６を用いて、キャリブレーション用画像上の座標（ｈｘ，ｈｙ）に変換する。そして、プロジェクタ内部のシステムコントローラは、キャリブレーション用画像上の座標（ｈｘ，ｈｙ）をコンピュータに対して送信する。

【００５３】

以上のように、第３の実施形態においては、デジタイザユニット及びプロジェクタにより座標入力システムを構成することができる。

【００５４】

以上に説明した実施形態においては、ＣＰＵ等によるソフトウェア的な処理によって実施しているが、他の実施形態として、ハードウェア的な処理によっても同じ機能を実現することが可能である。例えば、ＡＳＩＣ（Application Specific Integrated Circuit）等のハードウェアデバイスを用いることができる。

【００５５】

上述した実施形態によれば、キャリブレーションを行った結果から、座標の誤差の大きな矩形領域を自動的に判定して、誤差の大きな矩形領域に対してのみキャリブレーションを行うためのポイントを追加する。その結果、ユーザの作業負荷を抑えつつ、効率よく座標の誤差を低減させることが可能となる。

【００５６】

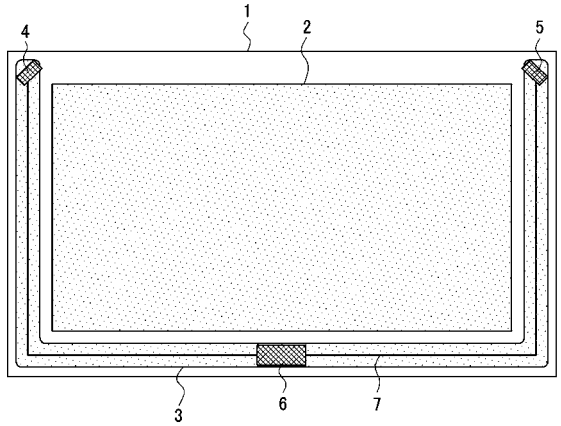
また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはＣＰＵやＭＰＵ等）がプログラムを読み出して実行する処理である。

【符号の説明】

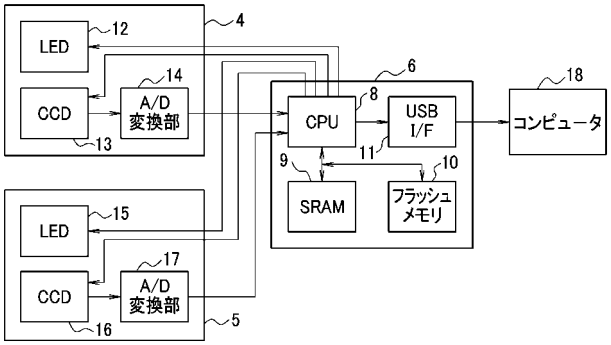
【００５７】

１：ホワイトボード、２：表示領域、３：デジタイザユニット、４、５：センサユニット、６：制御ユニット、７：再帰反射部、８：ＣＰＵ、９：ＳＲＡＭ、１０：フラッシュメモリ、１１：ＵＳＢインタフェース部、１２、１５：ＬＥＤ、１３、１６：ＣＣＤ、１４、１７：Ａ／Ｄ変換部、１８：コンピュータ

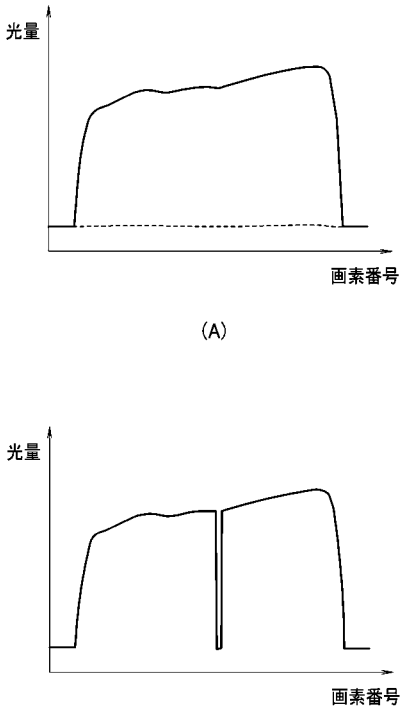
【図１】



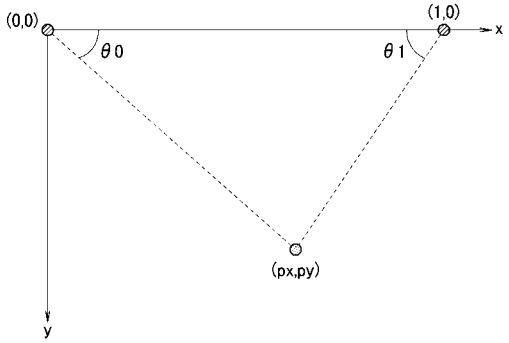
【図２】



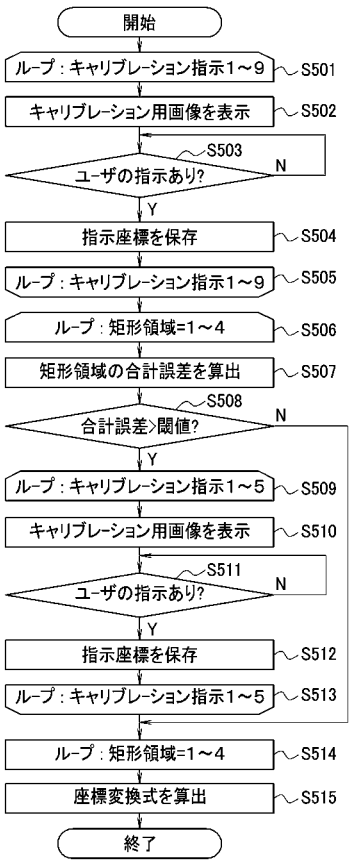
【図３】


光量
画素番号
(A)
光量
画素番号
(B)

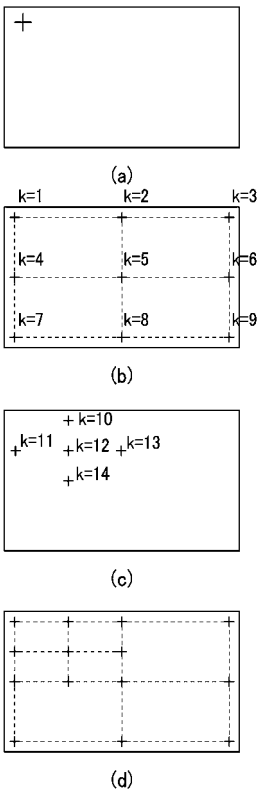
【図４】


(0,0)
(1,0)
x
θ0
θ1
(px,py)
y

【図５】


開始
ループ：キャリブレーション指示1～9
S501
キャリブレーション用画像を表示
S502
S503
ユーザの指示あり?
N
Y
指示座標を保存
S504
ループ：キャリブレーション指示1～9
S505
ループ：矩形領域=1～4
S506
矩形領域の合計誤差を算出
S507
S508
合計誤差>閾値?
N
Y
ループ：キャリブレーション指示1～5
S509
キャリブレーション用画像を表示
S510
S511
ユーザの指示あり?
N
Y
指示座標を保存
S512
ループ：キャリブレーション指示1～5
S513
ループ：矩形領域=1～4
S514
座標変換式を算出
S515
終了

【図６】


(a)
k=1
k=2
k=3
k=4
k=5
k=6
k=7
k=8
k=9
(b)
k=10
k=11
k=12
k=13
k=14
(c)
(d)

フロントページの続き

(51)Int.Cl.	FI			テーマコード（参考）
	Ｇ０９Ｇ	5/00	５５０Ｃ	
	Ｇ０９Ｇ	5/00	５３０Ｔ	
	Ｇ０９Ｇ	5/36	５２０Ｐ	
	Ｇ０９Ｇ	5/38	Ｚ	
	Ｇ０９Ｇ	5/00	５１０Ｂ	
	Ｇ０９Ｇ	5/00	５５０Ｄ	