(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3814333号
(P3814333)
(45)発行日　平成18年8月30日(2006.8.30)　(24)登録日　平成18年6月9日(2006.6.9)

(51)Int.Cl.　F I
G06F 3/041 (2006.01)　G06F 3/041 380N
G06F 3/041 380L

請求項の数 3　(全 17 頁)

(21)出願番号　特願平8-95616
(22)出願日　平成8年4月17日(1996.4.17)
(65)公開番号　特開平9-282079
(43)公開日　平成9年10月31日(1997.10.31)
審査請求日　平成15年3月14日(2003.3.14)

(73)特許権者　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　100076428
弁理士　大塚　康徳
(74)代理人　100093908
弁理士　松本　研一
(72)発明者　原田　隆史
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
(72)発明者　福田　亮治
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

審査官　圓道　浩史

最終頁に続く

(54)【発明の名称】情報処理方法とその装置

(57)【特許請求の範囲】

【請求項１】

入力面から所定距離内にペン入力デバイスが入っている状態と、その状態において前記ペン入力デバイスの第１位置座標を検知する第１検知手段と、

前記ペン入力デバイスが前記入力面に接触した入力状態と、その入力状態における前記ペン入力デバイスの第２位置座標を検知する第２検知手段と、

前記第１検知手段で検知された第１位置座標と、前記第１検知手段での検知後に前記第２検知手段で検知された第２位置座標との関係に基づいて、どの方向から前記入力状態になったかを判定する判定手段と、

前記判定手段の判定結果に対応するイベントの処理を実行させる実行手段とを備えることを特徴とする情報処理装置。

【請求項２】

前記入力面は、電磁誘導型タブレットによって構成されることを特徴とする請求項１に記載の情報処理装置。

【請求項３】

入力面から所定距離内にペン入力デバイスが入っている状態と、その状態において前記ペン入力デバイスの第１位置座標を検知する第１検知工程と、

前記ペン入力デバイスが前記入力面に接触した入力状態と、その入力状態における前記ペン入力デバイスの第２位置座標を検知する第２検知工程と、

前記第１検知工程で検知された第１位置座標と、前記第１検知工程での検知後に前記第

２検知工程で検知された第２位置座標との関係に基づいて、どの方向から前記入力状態になったかを判定する判定工程と、
　前記判定工程の判定結果に対応するイベントの処理を実行させる実行工程とを備えることを特徴とする情報処理方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、情報処理方法とその装置、特に、ペン入力デバイスを用いる情報処理方法とその装置に関する。
【０００２】
【従来の技術】
ＧＵＩの発達により、コンピュータ等の情報機器ではマウスを使って入力を行うシステム、あるいはアプリケーションが数多く開発されている。一方、携帯型の情報機器を中心にペン入力システムが開発され利用されている。これは、マウスを用いるにはマウスを動かすための場所が必要になるといった問題（省スペース）に対処するため、或はペン入力では画面に表示されているオブジェクトを直接操作できる、といった優れた操作性を持つという利点があるためである。ペン入力システムの中には、従来のマウスの替わりにペンを使うことの出来るものもある。このようなシステムにおいては、従来マウスを使って行っていた操作をペンで行えるようになっている。
【０００３】
【発明が解決しようとする課題】
マウスを使うシステムではキーボードが付属していることが前提となっており、マウスだけでなくキーボードも併用して入力を行うことがある。例えば、ＸＷｉｎｄｏｗシステムで、ターミナルアプリケーションソフトとして利用されるｘｔｅｒｍでは、“ｃｏｎｔｒｏｌ”キーとマウスボタンダウンにより、フォントサイズの変更等の端末属性の設定を行えるようになっている。
【０００４】
しかしながら、ペン入力機器では前記したように省スペースを考慮しているため、キーボードが付属しない場合が多い。この場合、マウスを使うシステムにおいて、マウスとキーボードを併用する操作をもつアプリケーションをペン入力システムではそのまま実行できないという問題が起こる。
これを解決するために、ペン入力システムではジェスチャーにより同様の操作ができるようになっているが、このジェスチャーの形状認識の際の誤認識、あるいは認識処理に時間がかかるといった点が問題となっている。
【０００５】
また、マウスとは違った操作を使用者は覚えなくてはならず、ユーザに負担をかけるという問題も起こる。
また、マウスシステムでのドラッグ操作（マウスのボタンを押しながら、マウスを移動する）に関していうと、マウスでは、使用中のマウスに装備されているボタンの数だけ違った操作を行うことができるが、ペン入力システムでは１種類しかできないといった問題が発生する。
【０００６】
また、ワークステーションの様に、複数のマウスボタンを備えている場合には、そのなかで最もよく使われるボタンが押されたときの操作をペンダウンに割り当てているため、２つ以上のボタンを備えたマウスを使うシステム、または、アプリケーションをサポートすることはできない。
また、ペンにスイッチが装備されている場合には、そのスイッチを押したままペンダウンした時に、その次によく使われるボタンが押されたときの操作を割り当てることにより、マウス利用時と同様の操作ができるものもある。
【０００７】

しかしながら、この場合でも３つ以上のボタンを備えたマウスに対応することはできない。

本発明は、上記従来例に鑑みてなされたもので、ペン入力デバイスを用いて、そのペン入力デバイスの簡単な動作に基づいて、対応する各コマンド処理を実行できる情報処理方法とその装置を提供することを目的とする。

【０００８】

【課題を解決するための手段】

上記の目的を達成するための本発明による情報処理装置は以下の構成を備える。すなわち、

入力面から所定距離内にペン入力デバイスが入っている状態と、その状態において前記ペン入力デバイスの第１位置座標を検知する第１検知手段と、

前記ペン入力デバイスが前記入力面に接触した入力状態と、その入力状態における前記ペン入力デバイスの第２位置座標を検知する第２検知手段と、

前記第１検知手段で検知された第１位置座標と、前記第１検知手段での検知後に前記第２検知手段で検知された第２位置座標との関係に基づいて、どの方向から前記入力状態になったかを判定する判定手段と、

前記判定手段の判定結果に対応するイベントの処理を実行させる実行手段とを備える。

【００１２】

また、上記の目的を達成するための本発明による情報処理方法は、

入力面から所定距離内にペン入力デバイスが入っている状態と、その状態において前記ペン入力デバイスの第１位置座標を検知する第１検知工程と、

前記ペン入力デバイスが前記入力面に接触した入力状態と、その入力状態における前記ペン入力デバイスの第２位置座標を検知する第２検知工程と、

前記第１検知工程で検知された第１位置座標と、前記第１検知工程での検知後に前記第２検知工程で検知された第２位置座標との関係に基づいて、どの方向から前記入力状態になったかを判定する判定工程と、

前記判定工程の判定結果に対応するイベントの処理を実行させる実行工程とを備える。

【００１９】

【発明の実施の形態】

はじめに、本発明の実施の形態の情報処理方法とその装置のポイントを要約した後に、その詳細な説明に入るものとする。

本発明の実施の形態の情報処理方法とその装置によれば、マウスなどのポインティング入力デバイスをサポートするアプリケーションやシステム側のプログラムをそのまま使用でき、マウスがサポートする操作入力をペン入力デバイスで置き換えることができる。

【００２０】

例えば、３つのボタンを有するマウスをサポートするワークステーションなどで、マウスの代わりにペン入力デバイスを用いる場合、ペン入力デバイスのペン先の動作方向を検出し、左側から入力された場合には、３つボタンマウスの左ボタンダウンに対応する操作を行えるようにし、真上からであれば真ん中のボタンダウン、右側からであれば右ボタンダウンに対応する処理を行う。こうすることにより、マウス操作環境で使われているアプリケーションを変更することなく、ペン操作環境で利用することができる。

【００２１】

以下、本発明の実施の形態の情報処理方法とその装置の詳細な説明を行う。

［第１の実施例］

図１に本実施の形態の情報処理装置の概要構成を示す。

ＣＰＵ２００は、本情報処理装置全体の制御を、メモリ２０２に格納された各種制御プログラムを読み出し、解釈し、実行することで行う。

【００２２】

ペン入力デバイス２０６は、ペンの座標とペンの状態（ステータス）を入力する。

キーボード２０３とポインティングデバイス２０４は、コマンドやデータの入力を行う。

ディスプレイモニタ２０１は、ＣＰＵ２００での処理結果や、キーボード２０３とポインティングデバイス２０４から入力したコマンドやデータを表示する。

【００２３】

タイマー２０５は、現在時刻をカウントし、ＣＰＵ２００の要請に基づいて、現在時刻を通知する。

図２に、図１の情報処理装置で処理される概要処理構成を示す。

図２を参照して、１は、ペンにより入力されたデータ（座標データとステータス）を受取るためのペン入力部である。２は、入力されたデータから入力属性を検出するための入力属性検出部である。入力属性は、ペン先の移動方向である。即ち、後述するが、ステータスが"プロキシミティー内"の座標値と"ペンダウン"の座標値によって、ペン先の移動方向を検出する。

【００２４】

３は、入力属性検出部２で検出された属性を基に、各マウスボタンに対応するイベントを発生させる、言い替えると、ペンイベントをマウスイベントに変換するイベント変換部である。

４は、イベント変換部から送られてきたイベントに対応する操作を実行するためのイベント実行部である。

【００２５】

これらの各部での処理は、プログラムとして、メモリ２０２に格納されており、ＣＰＵ２００によって実行される。

図２で処理の概要を述べたが、次に図３と図４を参照してより詳細な処理手順を説明する。

ここでは、ペン入力デバイスとして、電磁誘導型タブレットのように、ペン先がタブレット表面に付かない状態であっても座標を検出できるものを利用した場合について述べる。

【００２６】

このとき、入力デバイスからは座標値と共にその座標が検知されたとき、ペン先がタブレット面に付いているのか、それともペン先がタブレット面からある一定距離以内にあるのか、あるいは、ペン先がタブレット面からある一定距離以上離れているのかを示すデータが送られてくる。

以下では、ペン先がタブレット面からある一定距離以内にある場合には、

”ペン先がプロキシミティー内にある”

と呼び、ペン先がタブレット面からある一定距離以上離れている場合には、

”ペン先がプロキシミティー外にある”と呼ぶことにする。

【００２７】

一方、ペン先がタブレット面に付いている状態を、

”ペンダウン”

と呼ぶことにする。

以下では、マウスでは、マウスボタンを押しているときだけ、マウスボタン押下イベントが発生するとする。また、初期状態では、ペン先がプロキシミティー外にあるとする。ペン入力デバイスからは、ペン先がプロキシミティー外にある場合は、データは送られてこない。即ち、ペン先がプロキシミティー内にある場合とペンダウンの時に座標データをステータスが送られてくる。

【００２８】

尚、以下の説明では、処理で必要な各種変数名が、予告なしに現れるが、これらは、メモリ２０２にアサインされているものとする。

ステップＳ２０１では、ペン入力デバイスから座標値とステータス（上述の状態のうちの１つ）を取得する。

ステップＳ２０２では、ステータスのチェックを行い、ペン先がプロキシミティー内かどうかを判定する。プロキシミティー内であれば、ステップＳ２０３に進み、そうでなければ、ステップＳ２０１に戻る。

【００２９】
ステップＳ２０３では、ステップＳ２０１で入力した座標値とステータスをメモリ２０２に格納する。そして、ペン入力デバイスから次の座標値とステータスを入力する。
ステップＳ２０４では、ステップＳ２０３で入力したステータスをチェックし、ペンダウンかどうかの判定を行う。ペンダウンでなければ、ステップＳ２０５へ進み、入力したの入力座標データをメモリ２０２に格納する。そして、次の入力を得るためにステップＳ２０１に戻る。
【００３０】
また、ステップＳ２０４において、ペンダウンであると判定された場合には、ステップＳ２０８に進む。
ステップＳ２０８では、ペン先がどちらの方向から入力されたかを計算する。すなわち、ステップＳ２０３で格納されたペンダウンする前のプロキシミティー内での座標データと、ペンダウン時の座標データのＸ方向（横方向）の値を比較することにより、ペン先が左側からペンダウンしたのか、右側からペンダウンしたのかを判定する。
【００３１】
なお、この２点間のＸ方向の差が所定の閾値以下であれば、概上から入力したと判定する。従って、入力が３方向のいづれであるかを検知できる。
ステップＳ２０８で、左側から入力されたと判定された場合には、方向ステータスｄｓに"０"を設定する。また、概上ならば方向ステータスｄｓに"１"を設定する。また、右側からであれば、方向ステータスｄｓに"２"を設定する。
【００３２】
ステップＳ２０９では、ステップＳ２０８で得られた方向ステータスｄｓを、メモリ２０２０に保持する。
ステップＳ３０１では、方向ステータスｄｓをチェックして、"０"の場合にはステップＳ３０２に進む。また"１"の場合には、ステップＳ３０３に進む。また、"２"の場合には、ステップＳ３０４に進む。
【００３３】
ステップＳ３０２、ステップＳ３０３、ステップＳ３０４ではそれぞれ、左、中、右の各マウスボタン押下イベントを生成する。即ち、ステップＳ３０１、ステップＳ３０２、ステップＳ３０３、ステップＳ３０４の各処理により、ペン入力データがマウスイベントに変換される。
ステップＳ３０５では、入力したそれぞれのイベントに対応する所定の処理が実行される。ステップＳ３０５での処理は、アプリケーション側、または、オペレーティングシステム側で適応的に設定するものである。　ステップＳ３０５での処理が終了すると、次の入力を得るため、再びステップＳ２０１からの処理に戻り、同様の処理を繰り返す。
【００３４】
以上説明したように、本実施の形態のペン入力方式を用いることにより、電磁誘導型のような入力面にペンダウンしなくてもペン先の座標値が得られるペン入力デバイスを使った場合、ペンダウンする前の座標値とペンダウンした際の座標値を利用することにより、複数ボタンを備えたマウス利用時と同様の操作を、ペンでできるようになる。
［第２の実施の形態］
第２実施の形態では、入力面にペン先が触れたときに座標値が得られるペン入力デバイスを使った場合について、図５および図６を参照して説明する。
【００３５】
尚、このペン入力デバイスは、ペンダウンしたときと、ペンダウンした後は前回出力した座標からある一定距離以上離れたときに次の座標データを出力し、ペンアップしたときにその位置の座標データとステータスを出力する。
ここで、ステータスは、例えば、ペンダウン時には"１"、ペンアップ時には"０"となる。
【００３６】
また、以下の説明では、第１実施の形態と同様に、マウス側ではマウスボタンを押してい

るときだけ、マウスボタン押下イベントが発生するようなシステムに対応させる際の説明を行う。
以下、図５と図６を参照して、その構成に対応する詳細な処理を説明する。
ここで、処理のスタート時にはペンアップ状態にあるものとする。
【００３７】
ステップＳ４０１では、ペン入力デバイスからペンダウン入力があるかどうかを判定する。そして、入力がなければ、入力があるまで待つ。入力があった場合には、ステップＳ４０２に進む。
ステップＳ４０２では、現在の時刻をタイマー２０５から入力して、ステップＳ４０１で入力したペンダウン時の座標値とそのステータスと共に、メモリ２０２に格納する。
【００３８】
ステップＳ４０３では、ペン入力デバイスから次の入力を待つ。そして、入力があれば、ステップＳ４０４に進む。
ステップＳ４０４では、入力がペンアップかどうかを判断する。ペンアップであれば、ステップＳ４０１に戻る。そうでなければ、ステップＳ４０５に進む。
ステップＳ４０５では、現在の時刻をタイマー２０５から入力して、ステップＳ４０３で入力した座標値、ステータスと共に、メモリ２０２に格納する。
【００３９】
ステップＳ４０６では、ステップＳ４０２で保持した時刻とステップＳ４０５で保持した時刻の差を計算し、ステップＳ４０６において、その差が所定の閾値より大きかどうかを判定する。所定の閾値より小さい場合には、ステップＳ４０３に戻り、次の入力を得る。逆に、その閾値より大きい場合には、ステップＳ４０７に進む。
【００４０】
ステップＳ４０７では、ペン先が左側に動いたのか右側に動いたのかの方向を判定する。ここでは、ステップＳ４０２で保持した座標値のＸ方向（横方向）の値とステップＳ４０５で保持したＸ方向の値を比較することで、ペン先が左側に動いたのか右側に動いたのかを判定する。また、この２点間のＸ方向の差が所定の閾値以下であればペン先が動いていないものとして判定する。
【００４１】
そして、ペン先が左側に動いた場合には方向ステータスｄｓに"０"、動いていないときは"１"、右側に動いたときは"２"を設定する。
ステップＳ４０８では、この判定結果の方向ステータスｄｓをメモリ２０２に格納する。
ステップＳ５０１では、方向ステータスｄｓをチェックして、"０"の場合にはステップＳ５０２に、"１"の場合にはステップＳ５０３に、また、"２"の場合にはステップＳ５０４にそれぞれ分岐させる。
【００４２】
ステップＳ５０２、ステップＳ５０３、ステップＳ５０４ではそれぞれ、左、中、右の各マウスボタン押下イベントを生成する。即ち、ステップＳ５０１、ステップＳ５０２、ステップＳ５０３、ステップＳ５０４の各処理により、ペン入力データがマウスイベントに変換される。
ステップＳ５０５では、入力したそれぞれのイベントに対応する所定の処理が実行される。ステップＳ５０５での処理は、アプリケーション側、または、オペレーティングシステム側で適応的に設定するものである。
【００４３】
ステップＳ５０５での処理が終了すると、再び、ペン入力部１での処理となり、ステップＳ５０６からステップＳ５０８の処理を実行する。
ステップＳ５０６では、次の入力をペン入力デバイスから得るまで待つ。そして、入力があった場合、ステップＳ５０７に進む。
ステップＳ５０７では、入力したステータスがペンアップかどうかを判定する。そして、ペンアップであれば、再びステップＳ４０１から処理を開始する。そうでなければ、ステ

ップＳ５０８に進む。
【００４４】
ステップＳ５０８では、現在の方向ステータスｄｓとステップＳ５０６で得られた座標値とをイベント実行部に送り、ステップＳ５０１に再び戻る。
以上説明したように、本実施の形態のペン入力方式を用いることにより、入力面にペン先が触れたときにのみ座標値が得られるペン入力デバイスを使った場合にも、ペンダウン時の座標値と、ペンダウン後ある一定微小時間内に移動した座標値を利用することにより、複数ボタンを備えたマウス利用時と同様の操作をペンでできるようになる。
［第３の実施の形態］
第３の実施の形態は、上述の実施の形態とは別の方法を説明する。
本実施の形態のポイントは、ペン入力デバイスがペンダウンした位置を含む所定サイズの近傍領域を、ペンが何回通過するかをカウントして、マウスの各ボタンに対応させる処理を行うことである。
【００４５】
ここで、読者の理解を容易にするために、例えば、描画アプリケーションがマウス、キーボードを用いる場合、マウスのドラッグだけの場合にはマウスの動きに対応した自由曲線を描くモード、シフト（ｓｈｉｆｔ）キーを押しながら、所定の描画オブジェクト上からドラッグした場合には、描画オブジェクトのコピーとコピーしたオブジェクトの移動モードであるとする。
【００４６】
これらのモードを、ペン入力デバイスを用いて行うために、本実施の形態では、
ペンダウン後に入力するストロークがあらかじめ設定した大きさの入力開始点の近傍から出て、もう一度入力開始点の近傍を通過せずにペンアップされた場合にはそのストロークをそのまま表示するモードを備える。また、任意の描画オブジェクト上でペンダウンし、その後、ストロークが入力開始点の近傍から出てもう一度入力した開始点の近傍を通過し、その後、入力開始点の近傍を通過することなくウィンドウ中の任意の位置でペンアップした場合には、入力開始位置にある描画オブジェクトをペンアップした位置へ移動するモードを備える。
【００４７】
さらに、任意の描画オブジェクト上でペンダウンし、その後のストロークが入力開始点の近傍を２回通過し、ウィンドウ中の任意の位置でペンアップした場合には、入力開始位置にある描画オブジェクトをコピーし、ペンアップした位置へコピーされたオブジェクトを移動するモードを備える。
以上の処理モードを備えることで、マウス・キーボード環境で使われているアプリケーションを、ソースコード等の変更をすることなくペン操作環境で利用することができる。
【００４８】
また、入力開始点の近傍を通過した回数によって、イベントの区別を行うために、従来のペン入力デバイスのジェスチャーを入力して、その形状認識を行う処理に比べ、極めて処理が簡単なものとなる。
第３の実施の形態での処理を、図７、図８のフローチャートに基づいて説明する。
【００４９】
ステップＳ６０１では、各種変数名の初期化を行う。初期化項目としては、ストロークが入力開始点の近傍を通過した回数（以下、近傍通過数ｃｎｔと呼ぶ）と、前回の入力データが入力開始点の近傍領域内にあったかどうかを示すフラグ（以下、近傍内フラグｆｌｇと呼ぶ）の２項目であり、それぞれ"０"に設定する。
【００５０】
尚、上述の入力開始点の近傍とは、入力開始を中心とする正方形の領域であり、あらかじめその大きさは決められているが、ユーザによるメニュー操作によりその大きさを再設定することも可能である。
ステップＳ６０２では、ペン入力デバイスからのペンダウン入力を待つ。そして、入力の

あった位置に、表示中のオブジェクトが存在する場合には、ステップＳ６０３に進む。また、存在しない場合は、さらに、ペン入力デバイスからのペンダウン入力を待つ。

【００５１】

ステップＳ６０３では、その点を入力開始点として、その近傍領域の設定を行い、近傍内フラグをセットする。

ステップＳ６０４では、ユーザからの次のペンダウン移動入力（ペンダウンしたままでのペンの移動）あるいはペンアップ入力を待つ。そして、ペンダウン移動入力があった場合には、ステップＳ６０５に進む。また、ペンアップ入力があった場合には、ステップＳ６０１に戻る。

【００５２】

ステップＳ６０５では、ステップＳ６０４で得られた点の座標が、ステップＳ６０３で設定された近傍領域であるかどうかをチェックする。そして、その領域内であれば、ステップＳ６０６に進み、そうでなければ、ステップＳ７０２に進む。

ステップＳ６０６では、ステップＳ６０４で得られた点の座標が入力開始点の近傍内にあるかどうかチェックする。そして、近傍内にない場合には、ステップＳ６０７に進み、近傍内フラグをリセットし、ステップＳ６０４に戻り、次の入力を待つ。

【００５３】

一方、ステップＳ６０６で、近傍内にあると判断された場合には、ステップＳ６０８に進む。

ステップＳ６０８では、近傍内フラグがセットされているかどうかを調べ、セットされていなければ、ステップＳ６０９に進む。また、セットされていれば、ステップＳ６０４に戻る。

【００５４】

ステップＳ６０９では、近傍通過数ｃｎｔを１カウントアップする。

ステップＳ６１０では、近傍内フラグをセットする。そして、ステップＳ６０４に戻り、次の入力を待つ。

ステップＳ７０１では、

ステップＳ７０２では、近傍通過数ｃｎｔをチェックして、"０"の場合にはステップＳ７０３に進む。また、"１"の場合には、ステップＳ７０４に進む。また、"２"の場合には、ステップＳ７０５に進む。

【００５５】

ステップＳ７０３では、ボタン押下イベントを生成する。

ステップＳ７０４では、コントロール（ｃｏｎｔｒｏｌ）キー押下イベント＋マウスボタン押下イベントを生成する。

ステップＳ７０５では、シフト（ｓｈｉｆｔ）キー押下イベント＋マウスボタン押下イベントを生成する。

【００５６】

このように、ステップＳ７０２からＳ７０５の各処理により、ペン入力データがマウス＋キーボードイベントに変換される。

こうして変換されたイベントは、ステップＳ７０６で、そのイベントに対応する各処理が実行される。

ステップＳ７０７では、新たにペン入力があるかどうかをチェックする。ペンアップを入力すると、ステップＳ６０１に戻る。また、そうでない場合には、ステップＳ７０２にもどり、得られた入力点の座標値と保持している近傍通過数をもとに処理を続ける。

【００５７】

なお、ステップＳ７０３、ステップＳ７０４、ステップＳ７０５のいずれかで生成される各種イベントは、例えば、図９に示すように予め用意された各デバイス間のイベント対応テーブル（メモリ２０２に格納されている）を、近傍通過数ｃｎｔで引くことにより、変換先のイベントを得ることができる。

尚、イベント対応テーブルは、ユーザによるメニュー操作により変更、追加、削除するこ

とができる。
【００５８】
次に、図１０に、本実施の形態のペン入力方法での画面操作例を示す。
これは、マウスとキーボードを備える情報処理装置側で、ｓｈｉｆｔキーを押下したままマウスドラッグすることで、操作対象オブジェクトのコピーと移動を行う場合での操作例を示す。この例では、近傍通過数が１のときシフト（ｓｈｉｆｔ）キー押下イベントとマウスボタン押下イベントに変換される。
【００５９】
以上、説明したように、本実施の形態のペン入力方法をもちいると、入力したストロークが入力開始点を中心とした、あらかじめ設定された多きさの領域内からでるまでに入力開始点の近傍を通過した回数により識別されるイベントを生成することができる。そして、このイベントをマウスとキーボードを併用する操作に割り当てることで、従来のマウス、キーボードを使うことを前提に作られたアプリケーションあるいはシステムを、ソースコード等の変更をすることなくペン入力に基づいてできるようになる。
【００６０】
（第４の実施例）
第４実施例では、入力されたストロークに関する時間情報をさらに用いて、各イベントを生成する処理方法を図１１を用いて説明する。
本実施の形態での処理のポイントは、ペン入力デバイスからの入力ストロークが、一旦入力開始点の近傍を離れ、次に、ストロークが入力開始点の近傍に入るまでの時間が所定時間以内ならば通過と判定し、そうでなければ、それまでの通過した回数を入力属性として検出することにある。
【００６１】
図１１を参照して、ステップＳ８０１では、以下の変数名の初期化を行う。即ち、ストロークが入力開始点の近傍を通過した回数（以下、"近傍通過数ｐｃｎｔ"と呼ぶ）と、前回の入力データが入力開始点の近傍領域内にあったかどうかを示すフラグ（以下、"近傍内フラグｎｆｌｇ"と呼ぶ）をそれぞれ"０"に初期化する。　なお、入力開始点の近傍とは、入力開始点を中心とする所定のサイズを有する正方形の領域である。このサイズは、ユーザによるメニュー操作により再設定することができる。
【００６２】
また、近傍通過数ｐｃｎｔのカウントアップは、本実施の形態では、現在の時刻と前回入力開始点の近傍にあった時刻との時間差が所定の閾値以内であるかどうかに基づいて行う。この所定の閾値は、ユーザによるメニュー操作により再設定することができる。
ステップＳ８０２では、ペンダウン入力を待ち、ペンダウン入力があった位置にオブジェクトが存在する場合には、ステップＳ８０３に進む。ここで、オブジェクトとは、表示されている画像上でのオブジェクトであり、このオブジェクトに関するディスプレイ２０１上での位置情報を含む各情報は、メモリ２０２に格納されている。そして、この位置情報とペンダウン入力の位置を比較することで、ペンダウン入力があった位置にオブジェクトが存在するかどうかを判断できる。
【００６３】
ステップＳ８０３では、そのペンダウン位置を入力開始点として、その近傍領域の設定を行い、また、近傍内フラグｎｆｌｇをセットする。
ステップＳ６０４では、タイマー２０５から現在の時刻を取得し、メモリ２０２に保持する（以下、この時刻を基準時刻と呼ぶ）。
ステップＳ８０５では、次のペンダウン移動入力（ペンダウンしたままの移動）を待ち、この入力があった場合には、ステップＳ８０６に進む。
なお、ステップＳ８０５においてペンアップされた場合にはステップＳ８０１に戻る。
ステップＳ８０６では、現在の時刻を取得する。
【００６４】
ステップＳ８０７では、ステップＳ８０４とステップＳ８０６で取得した時刻を比較し、

その差が所定の閾値より大きいかどうかを判断する。大きい場合は、第３の実施の形態でのイベント変換処理と同じ処理を行う。即ち、図８のステップＳ７０２からの処理と同様の処理を行うので、ここでは、説明の重複を避けるため説明を省略する。また、大きくないならば、ステップＳ８０８に進む。

【００６５】

ステップＳ８０８では、現在の入力点が入力開始点の近傍かどうか調べる。近傍内でなければ、ステップＳ８０９に進む。近傍内であれば、ステップＳ８１２に進む。

ステップＳ８０９では、近傍内フラグｎｆｌｇがセットされているかどうかを調べる。そして、セットされていなければ、ステップＳ８０５に戻る。逆に、セットされていれば、ステップＳ８１０に進む。

【００６６】

ステップＳ８１０では、タイマー２０５から現在の時刻を入力し、基準時刻としてメモリ２０２に保持する。

ステップＳ８１１では、近傍内フラグｎｆｌｇをリセットする。そして、ステップＳ８０５に戻る。

ステップＳ８０８で近傍内であると判断された場合には、ステップＳ８１２で近傍内フラグｎｆｌｇがセットされているかどうかを調べる。そして、セットされていれば、ステップＳ８０５に戻り、そうでなければ、ステップＳ８１３に進み、近傍通過数ｐｃｎｔを"１"カウントアップする。

【００６７】

ステップＳ８１４では、近傍内フラグｎｆｌｇをセットする。そして、ステップＳ８０５に戻り、次の入力を待つ。

以上説明したように、本実施の形態のペン入力方法を用いると、ストロークが一旦、入力開始点の近傍を離れ、次に、ストロークが入力開始点の近傍に入るまでの時間が一定時間以内ならば通過と判定し、そうでなければ、それまでの通過した回数を入力属性として確定する。そして、その確定回数に基づいて、対応するイベント、例えば、マウスとキーボードを併用する所定の操作を割り当てる。このことで、従来のマウス、キーボードの代わりに、ペンだけでできるようになる。

【００６８】

なお、本実施の形態では、ストロークが一旦入力開始点の近傍を離れ、次に、ストロークが入力開始点の近傍に入るまでの時間が所定時間以内ならば通過とし、そうでなければ、それまでの通過回数を入力属性として検出したが、図１１のステップＳ８１１での処理を行わなければ、入力開始時点からの一定時間内に入力開始点近傍を通過した回数を入力属性として利用することが可能である。

【００６９】

また、第３実施の形態、第４実施の形態において説明した入力属性検出方式を組み合わせたものを入力属性検出方式とすることでも、同様の効果が得られる。

（第５実施の形態）

本発明の第５の実施の形態の処理構成の概要を図１２に示す。

図１２の７１はペンにより入力されたデータを受取るためのペン入力部である。また、７２は入力されたストロークから入力属性を検出するための入力属性検出部である。なお、本実施の形態では、入力属性として、入力したストロークが入力開始点を中心とした所定の大きさの領域内から出るまでに入力開始点の近傍を通過した回数を用いる。

【００７０】

７３は、入力属性検出部７２で検出された属性に基づいて、対応するイベントを実行するためのイベント実行部である。

図１３は、イベント実行部７３（図１２）での処理のフローチャート示す。

また、図１４は、イベント実行部７３で保持される近傍通過数とコマンドの対応をそれぞれ示す。

【００７１】

以下、図１３を参照して、イベント実行処理を説明する。なお、ペン入力、入力属性検出処理の流れは、第３実施の形態と同様である。即ち、図７の各ステップを実行して、ステップＳ６０５での分岐条件が成立すると、ステップＳ９０２（図１３）にエントリする。ステップＳ９０２では、入力属性検出部２で生成された近傍通過数をチェックして、その数に基づいて、ステップＳ８０３、ステップＳ９０４、ステップＳ９０５の各ステップに分岐する。これらのステップでは、図９の示す近傍通過数とコマンドの対応に従い、近傍通過数に対応したコマンドを生成する。

【００７２】

ステップＳ９０２では、近傍通過数が"０"の場合にはステップＳ８０３に、"１"の場合にはステップＳ８０４に、また"２"の場合にはステップＳ８０５にそれぞれ分岐する。

ステップＳ８０３では、削除コマンドを生成する。

そして、ステップＳ８０６では、次の入力を待つ。そして、入力が無くなった時点で削除コマンドを実行し（ステップＳ８０７）、図７のステップＳ６０１へ進む。

【００７３】

一方、ステップＳ８０４では、移動コマンドを生成する。

次に、ステップＳ９０８では、移動コマンドと現在の座標値を基に、その移動コマンドを実行する。

ステップＳ８０９では、入力を待ち、ペンアップされるまで移動コマンドを繰り返す。そして、図７のステップＳ６０１へ進む。

【００７４】

他方、ステップＳ８０５では、コピーコマンドを生成する。

次に、ステップＳ８１０では、近傍通過数を"１"に設定する。そして、ステップＳ８０８に進み、コピーコマンドを実行する。そして、ステップＳ８０９で次の入力を待ち、ペンムーブ（Ｐｅｎ　ｍｏｖｅ）であれば、ペンアップされるまで移動コマンドを実行する。

【００７５】

尚、ステップＳ８０７、ステップＳ８０８でのコマンドの実行に関しては、本方法をインプリメントするシステム側で通常用意されるものであるので、ここでは詳しい説明は省略する。

以上説明したように、本実施の形態のペン入力方式を用いると、入力したストロークが入力開始点を中心とした、あらかじめ設定された大きさの領域内から出るまでに入力開始点の近傍を通過した回数により、システムあるいはアプリケーションで用意されるコマンドを割り当てることで、従来のシステムあるいはアプリケーションを、ソースコード等の変更をすることなくペンだけで操作できるようになるだけでなく、ジュエスチャ入力に基づく形状認識のため起こっていた誤認識、認識時間が長いという問題を解決することができる。

【００７６】

なお、本発明は、複数の機器から構成されるシステムに適用しても、一つの機器からなる装置に適用してもよい。

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。

【００７７】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク，ハードディスク，光ディスク，光磁気ディスク，ＣＤ－ＲＯＭ，ＣＤ－Ｒ，磁気テープ，不揮発性のメモリカード，ＲＯＭなどを用いることができる。

【００７８】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【００７９】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【００８０】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図１５のメモリマップ例に示す上述の各フローチャートの処理に対応する各モジュールを記憶媒体に格納することになる。

すなわち、少なくとも、ペンの状態と座標を入力する「ペン入力処理モジュール」、上述したように動的なペンの移動に伴う座標値の移動とその時刻等に基づいて、イベントを識別する「入力属性検出モジュール」、識別されたイベントをマウスなどのイベントに変換する「イベント変換モジュール」、変換されたイベントに対応するコマンドを実行する「コマンド実行モジュール」の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【００８１】

以上説明したように、本実施の形態によれば、マウスとキーボードを使うシステムにおいて、マウスとキーボードを同時に使うような操作を持つアプリケーションでも、ソースコードを変更することなく、そのままペン操作環境で利用することができる。

また、ペンだけで操作できる環境を提供できるだけでなく、従来、ジェスチャーの形状認識の際に起こっていた誤認識、認識時間にかかる時間等の問題を解決することができる。

【００８２】

また、使用者も新しい操作を覚える必要はなく、従来のマウス操作と同じ間隔感覚で使用できる。

【００８３】

【発明の効果】

以上説明したように本発明によれば、ペン入力デバイスを用いて、そのペン入力デバイスの簡単な動作に基づいて、対応する各コマンド処理を実行できる。

【図面の簡単な説明】

【図１】本発明に係る実施の形態の情報処理装置の基本構成例を示す図である。

【図２】第１の実施の形態の処理構成の概要を説明するための図である。

【図３】第１の実施の形態における処理の流れを示すフローチャートである。

【図４】　第１の実施の形態における処理の流れを示すフローチャートである。

【図５】第２の実施の形態における処理の流れを示すフローチャートである。

【図６】　第２の実施の形態における処理の流れを示すフローチャートである。

【図７】第３の実施の形態における処理の流れを示すフローチャートである。

【図８】　第３の実施の形態における処理の流れを示すフローチャートである。

【図９】イベント変換処理で参照されるイベント変換対応表である。

【図１０】表示画面でペン入力を行った軌跡の例を示す図である。

【図１１】第４の実施の形態の処理の流れを示すフローチャートである。

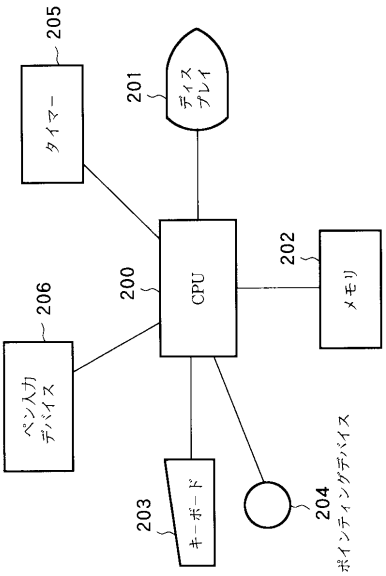
【図１２】第５の実施の形態の処理概要を説明するための図である。

【図１３】第５の実施の形態の処理のフローチャートである。

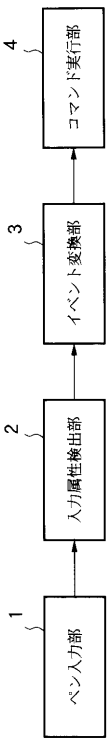
【図１４】近傍通過数とコマンドの対応を示す図である。

【図１５】コンピューター読み取り可能な所定の記録媒体に格納された各プログラムモジュールのレイアウトを示す図である。

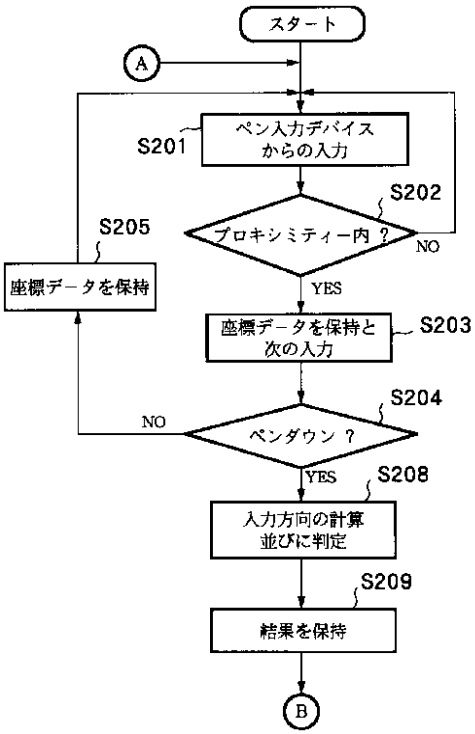
【図１】



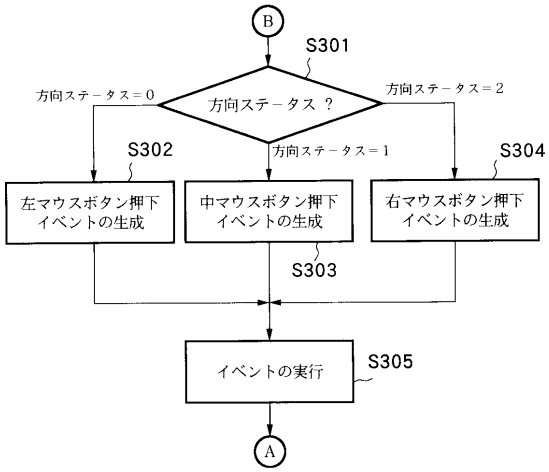
【図２】



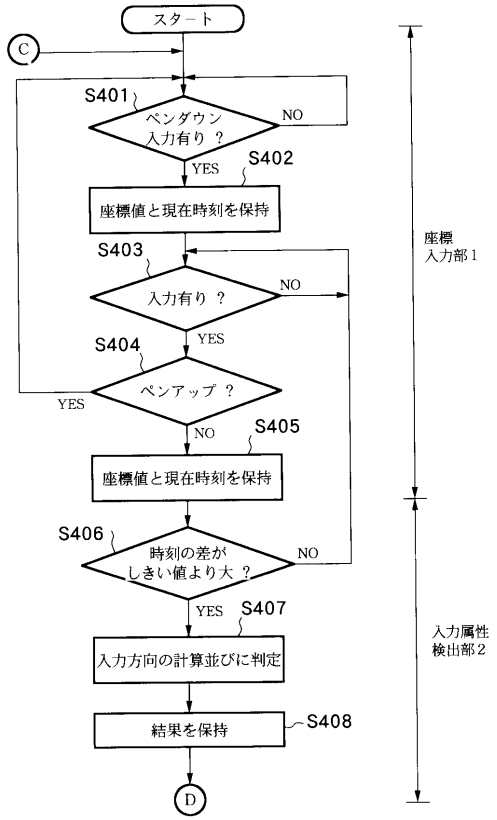
【図３】


スタート
A
S201
ペン入力デバイス
からの入力
S202
プロキシミティー内 ？
NO
YES
S205
座標データを保持
座標データを保持と
次の入力
S203
S204
NO
ペンダウン ？
YES
S208
入力方向の計算
並びに判定
S209
結果を保持
B

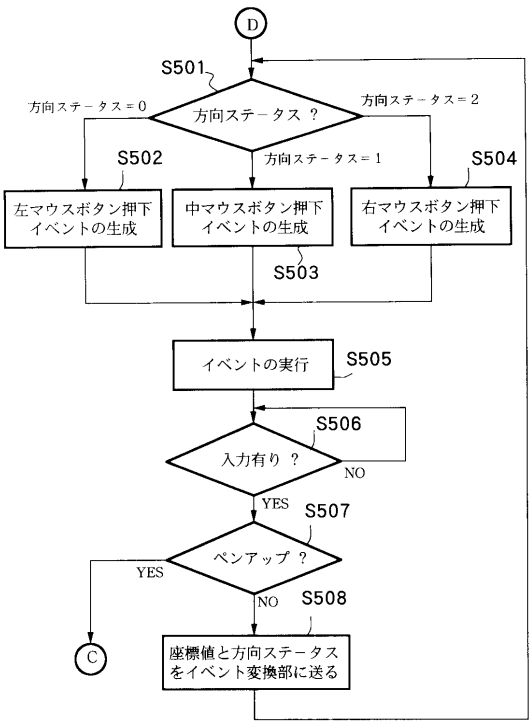
【図４】


B
S301
方向ステータス ？
方向ステータス＝0
方向ステータス＝1
方向ステータス＝2
S302
S304
左マウスボタン押下
イベントの生成
中マウスボタン押下
イベントの生成
右マウスボタン押下
イベントの生成
S303
イベントの実行
S305
A

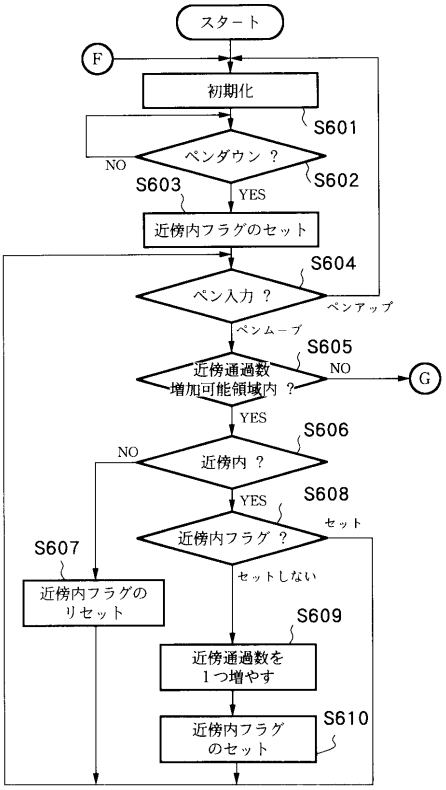
【図５】


スタート
C
S401
ペンダウン
入力有り ？
NO
YES
S402
座標値と現在時刻を保持
S403
入力有り ？
NO
YES
S404
ペンアップ ？
YES
NO
S405
座標値と現在時刻を保持
S406
時刻の差が
しきい値より大 ？
NO
YES
S407
入力方向の計算並びに判定
結果を保持
S408
D
座標
入力部１
入力属性
検出部２

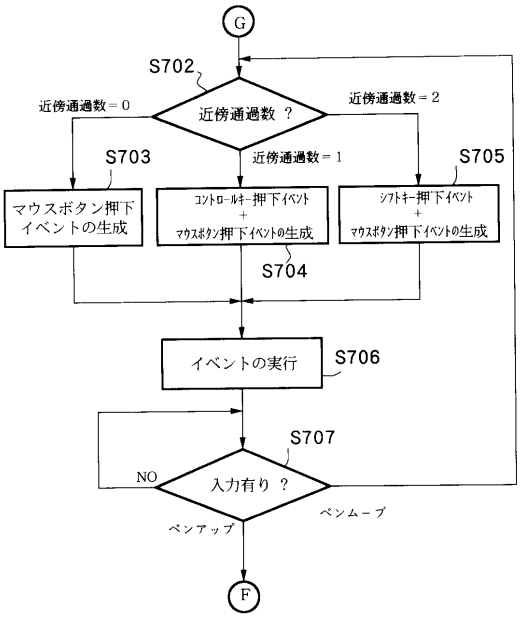
【図６】


D
S501
方向ステータス ？
方向ステータス＝0
方向ステータス＝1
方向ステータス＝2
S502
S504
左マウスボタン押下
イベントの生成
中マウスボタン押下
イベントの生成
右マウスボタン押下
イベントの生成
S503
イベントの実行
S505
S506
入力有り ？
NO
YES
S507
ペンアップ ？
YES
NO
S508
C
座標値と方向ステータス
をイベント変換部に送る

【図７】



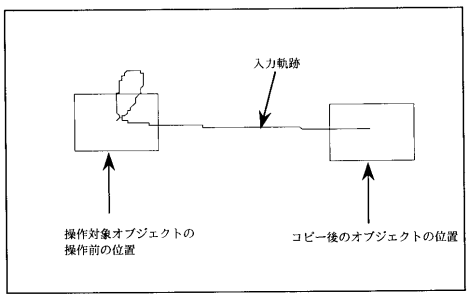
【図８】



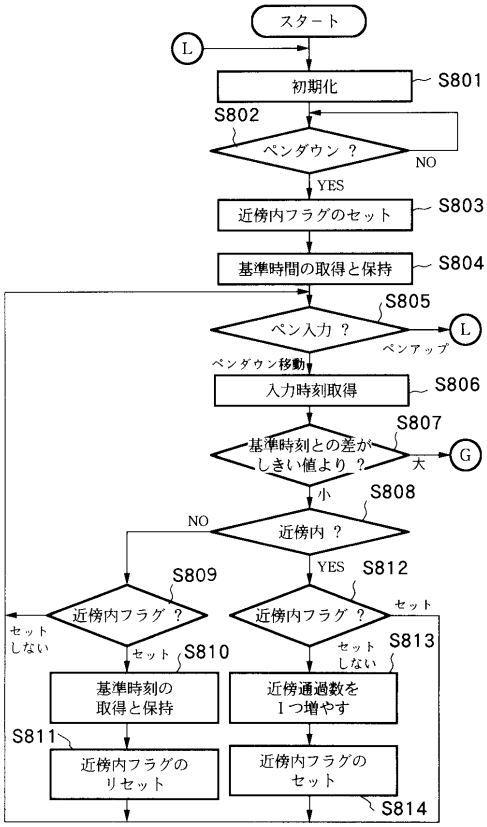
【図９】

近傍通過数	キーボードイベント	マウスイベント
0		left
1	control	left
2	shift	left
3		middle
4	middle	middle
5		right

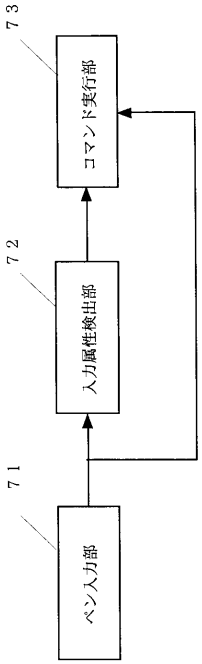
【図１０】



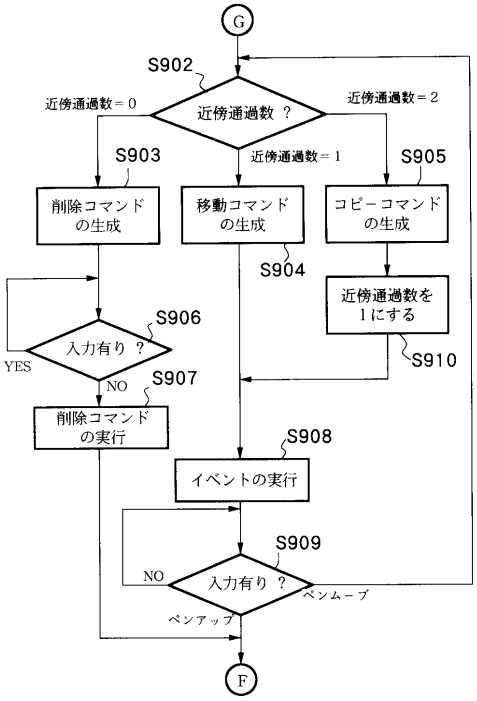
【図１１】



【図１２】



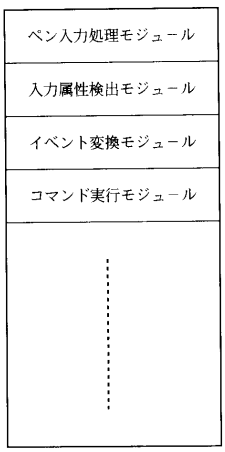
【図１３】



【図１４】

近傍通過数	生成コマンド
0	delete
1	move
2	copy

【図１５】



フロントページの続き

(56)参考文献 特開平０７－２００１２６（ＪＰ，Ａ）
特開平０７－３３４２９５（ＪＰ，Ａ）
特開平０６－２８９９８２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 3/041