

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5468005号
(P5468005)

(45) 発行日 平成26年4月9日(2014.4.9)

(24) 登録日 平成26年2月7日(2014.2.7)

(51) Int.Cl.

F I

G 0 6 F 3/048 (2013.01)

G 0 6 F 3/048 6 5 1 A

G 0 6 F 3/14 (2006.01)

G 0 6 F 3/14 3 1 0 A

請求項の数 49 (全 27 頁)

(21) 出願番号 特願2010-524305 (P2010-524305)
 (86) (22) 出願日 平成20年9月11日(2008.9.11)
 (65) 公表番号 特表2010-539566 (P2010-539566A)
 (43) 公表日 平成22年12月16日(2010.12.16)
 (86) 国際出願番号 PCT/AU2008/001342
 (87) 国際公開番号 W02009/033216
 (87) 国際公開日 平成21年3月19日(2009.3.19)
 審査請求日 平成23年9月12日(2011.9.12)
 (31) 優先権主張番号 2007904925
 (32) 優先日 平成19年9月11日(2007.9.11)
 (33) 優先権主張国 オーストラリア(AU)

(73) 特許権者 510068105
 スマート・インターネット・テクノロジー
 ・シーアールシー・プロプライエタリー・
 リミテッド
 SMART INTERNET TECH
 NOLOGY CRC PTY LTD
 オーストラリア2015ニューサウスウェ
 ールズ州イブリー、ロコモティブ・ストリ
 ート2番、スウィート9003、ロコモテ
 イブ・ワークショップ、オーストラリア・
 テクノロジー・パーク
 (74) 代理人 100101454
 弁理士 山田 卓二
 (74) 代理人 100081422
 弁理士 田中 光雄

最終頁に続く

(54) 【発明の名称】 コンピュータインターフェースのためのインターフェースエレメント

(57) 【特許請求の範囲】

【請求項 1】

コンピューティングインターフェース上のオブジェクトを操作する方法であって、
 上記方法は、

上記コンピューティングインターフェース内に影響の領域を画成し、上記影響の領域の
 内部に表示されるオブジェクトのサイズを変更するインターフェースエレメントを表示す
 ることと、

上記表示されるオブジェクトと上記インターフェースエレメントに関連する上記影響の
 領域との間の距離を変更するユーザ入力を受信することと、

上記表示されるオブジェクトのサイズを上記表示されるオブジェクトと上記影響の領域
 の中心との間の距離が減少するにつれて操作することとを含み、

上記操作は上記表示されるオブジェクトのサイズを縮小させることを含む方法。

【請求項 2】

上記ユーザ入力が上記表示されるオブジェクトを上記インターフェースエレメントに関
 連する上記影響の領域内に移動させるとき、上記表示されるオブジェクトのサイズを縮小
 させることを含む請求項 1 記載の方法。

【請求項 3】

上記ユーザ入力が上記インターフェースエレメントを操作して上記インターフェースエ
 レメントに関連する上記影響の領域の内部に上記オブジェクトが表示されるとき、上記表
 示されるオブジェクトのサイズを縮小させることを含む請求項 1 記載の方法。

10

20

【請求項 4】

上記インターフェースエレメントの上記影響の領域は、上記インターフェースエレメントを移動させることによって操作される請求項 3 記載の方法。

【請求項 5】

上記インターフェースエレメントの上記影響の領域は、上記インターフェースエレメントをサイズ変更することによって操作される請求項 3 記載の方法。

【請求項 6】

上記ユーザ入力が入力が上記インターフェースエレメントの動作を停止させるとき、上記表示されるオブジェクトのサイズを操作前のサイズに戻すことを含む請求項 1 記載の方法。

【請求項 7】

上記表示されるオブジェクトのサイズは、上記影響の領域を通る移動の方向にしたがって変化させられる請求項 1 から 6 のうちのいずれか 1 つの請求項記載の方法。

【請求項 8】

上記方向は、半径方向である請求項 7 記載の方法。

【請求項 9】

上記ユーザ入力が入力がオブジェクトと上記影響の領域の中心との間の距離を増加させるとき、上記表示されるオブジェクトのサイズを拡大させることを含む請求項 1 から 8 までのうちのいずれか 1 つの請求項記載の方法。

【請求項 10】

上記方法は、複数の表示されるオブジェクトを上記受信された入力の性質にしたがって操作することを含む請求項 1 から 9 のうちのいずれか 1 つの請求項記載の方法。

【請求項 11】

上記影響の領域の中へ、上記影響の領域の外へ、又は上記影響の領域を通して移動するすべての表示されるオブジェクトのサイズを変化させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 12】

上記インターフェースエレメントが移動されるときに、上記インターフェースエレメントに関連する上記影響の領域の内部に位置するすべての表示されるオブジェクトを移動させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 13】

上記ユーザ入力が入力が、上記表示されるオブジェクトを、上記インターフェースエレメントが入ることを防止されるゾーンである上記コンピューティングインターフェースの禁止されたゾーンの中に移動させるとき、上記インターフェースエレメントに関連する上記影響の領域の内部に位置する上記表示されるオブジェクトを、上記インターフェースエレメントから分離させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 14】

上記ユーザ入力が入力が上記インターフェースエレメントのアトラクション機能を起動させるとき、上記インターフェースエレメントに関連する上記影響の領域内のすべての表示されるオブジェクトを、上記影響の領域の中心に向かって移動させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 15】

上記ユーザ入力が入力が上記インターフェースエレメントのアトラクション機能を起動させるとき、上記コンピューティングインターフェース内のすべての表示されるオブジェクトを、上記インターフェースエレメントに関連する上記影響の領域の中心に向かって移動させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 16】

上記ユーザ入力が入力が上記インターフェースエレメントのアトラクション機能を起動させるとき、選択された属性を有する表示されるオブジェクトのサブセットを上記インターフェースエレメントに関連する上記影響の領域の中心に向かって移動させることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

10

20

30

40

50

【請求項 17】

上記インターフェースエレメントに関連する上記影響の領域の中心に到達する表示されるオブジェクトを隠すことを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 18】

上記受信された入力が入力インターフェースエレメントの動作を停止させるとき、上記インターフェースエレメントに関連する上記影響の領域の内部のすべての表示されるオブジェクトを上記影響の領域の中心から遠ざけることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 19】

上記受信された入力が入力インターフェースエレメントの反発機能を起動させるとき、上記コンピューティングインターフェース内のすべての表示されるオブジェクトを上記インターフェースエレメントに関連する上記影響の領域の中心から遠ざけることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 20】

上記受信された入力が入力インターフェースエレメントの反発機能を起動させるとき、選択された属性を有する表示されるオブジェクトのサブセットを上記インターフェースエレメントに関連する上記影響の領域の中心から遠ざけることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 21】

上記受信された入力が入力インターフェースエレメントの削除機能を起動させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の表示されるオブジェクトを削除することを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 22】

上記受信された入力が入力インターフェースエレメントをサイズ変更させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の表示されるオブジェクトをサイズ変更することを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 23】

上記受信された入力が入力インターフェースエレメントの動作を停止させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の隠された表示されるオブジェクトを見せることを含む請求項 1 から 10 のうちのいずれか 1 つの請求項記載の方法。

【請求項 24】

コンピューティングインターフェース上のオブジェクトを操作するコンピュータプログラムを記憶するコンピュータ読み取り可能な記憶媒体であって、

上記コンピュータプログラムは、コンピュータによって実行されるときに、上記コンピュータに、

上記コンピューティングインターフェース内に影響の領域を画成し、上記影響の領域の内部に表示されるオブジェクトのサイズを変更するインターフェースエレメントを表示させ、

上記表示されるオブジェクトと上記インターフェースエレメントに関連する上記影響の領域との間の距離を変更するユーザ入力を受信させ、

上記表示されるオブジェクトのサイズを上記表示されるオブジェクトと上記影響の領域の中心との間の距離が減少するにつれて操作させる命令を備え、

上記操作は上記表示されるオブジェクトのサイズを縮小させることを含むコンピュータ読み取り可能な記憶媒体。

【請求項 25】

コンピューティングインターフェース上のオブジェクトを操作する装置であって、

上記装置は、

コンピューティングインターフェースを表示するように設けられた表示エリアを有するディスプレイと、

10

20

30

40

50

入力装置と、

プロセッサとを備え、

上記プロセッサは、

上記表示エリア内に影響の領域を画成し、上記影響の領域の内部に表示されるオブジェクトのサイズを変更するインターフェースエレメントを表示させ、

表示されるオブジェクトと上記インターフェースエレメントの上記影響の領域との間の距離を変更させるユーザ入力を上記入力装置から受信し、

上記表示されるオブジェクトのサイズを、上記表示されるオブジェクトと上記影響の領域の中心との間の距離が減少するにつれて操作するように構成され、

上記操作は上記表示されるオブジェクトのサイズを縮小させることを含む装置。

10

【請求項 26】

コンピューティングインターフェース上のオブジェクトを操作する装置であって、

上記装置は、

コンピューティングインターフェースを表示するように設けられた表示エリアを有するディスプレイと、

入力装置と、

上記コンピューティングインターフェース内に影響の領域を画成し、上記影響の領域の内部に表示されるオブジェクトのサイズを変更するインターフェースエレメントを表示する手段と、

表示されるオブジェクトと上記インターフェースエレメントに関連する上記影響の領域との間の距離を変更するユーザ入力を上記入力装置から受信する手段と、

20

上記表示されるオブジェクトのサイズを上記表示されるオブジェクトと上記影響の領域の中心との間の距離が増加するにつれて操作する手段とを備え、

上記操作は上記表示されるオブジェクトのサイズを拡大させることを含む装置。

【請求項 27】

コンピューティングインターフェース上のオブジェクトを操作する方法であって、

上記方法は、

上記コンピューティングインターフェース内に影響の領域を画成し、上記影響の領域の内部に表示されるオブジェクトのサイズを変更するインターフェースエレメントを表示することと、

30

上記表示されるオブジェクトと上記インターフェースエレメントに関連する上記影響の領域との間の距離を変更するユーザ入力を受信することと、

上記表示されるオブジェクトのサイズを上記表示されるオブジェクトと上記影響の領域の中心との間の距離が増加するにつれて操作することとを含み、

上記操作は上記表示されるオブジェクトのサイズを拡大させることを含む方法。

【請求項 28】

上記ユーザ入力が上記表示されるオブジェクトを上記インターフェースエレメントに関連する上記影響の領域の中心から遠ざけると、上記表示されるオブジェクトのサイズを拡大させることを含む請求項 27 記載の方法。

【請求項 29】

40

上記ユーザ入力が上記インターフェースエレメントに関連する上記影響の領域を操作して上記影響の領域の中心と上記表示されるオブジェクトとの間の距離を増加させるとき、上記表示されるオブジェクトのサイズを拡大させることを含む請求項 27 記載の方法。

【請求項 30】

上記インターフェースエレメントの上記影響の領域は、上記インターフェースエレメントを移動させることによって操作される請求項 29 記載の方法。

【請求項 31】

上記インターフェースエレメントの上記影響の領域は、上記インターフェースエレメントをサイズ変更することによって操作される請求項 29 記載の方法。

【請求項 32】

50

上記ユーザ入力が入力されたインターフェースエレメントの動作を停止させるとき、上記表示されるオブジェクトのサイズを操作前のサイズに戻すことを含む請求項 27 記載の方法。

【請求項 33】

上記表示されるオブジェクトのサイズは、上記影響の領域を通る移動の方向にしたがって変化させられる請求項 27 から 32 のうちのいずれか 1 つの請求項記載の方法。

【請求項 34】

上記方向は、半径方向である請求項 33 記載の方法。

【請求項 35】

上記ユーザ入力が入力されたオブジェクトと上記影響の領域の中心との間の距離を減少させるとき、上記表示されるオブジェクトのサイズを縮小させることを含む請求項 27 から 33 までのうちのいずれか 1 つの請求項記載の方法。

10

【請求項 36】

上記方法は、複数の表示されるオブジェクトを上記受信された入力の性質にしたがって操作することを含む請求項 27 から 35 のうちのいずれか 1 つの請求項記載の方法。

【請求項 37】

上記影響の領域の中へ、上記影響の領域の外へ、又は上記影響の領域を通して移動するすべての表示されるオブジェクトのサイズを変化させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 38】

上記インターフェースエレメントが移動されるときに、上記インターフェースエレメントに関連する上記影響の領域の内部に位置するすべての表示されるオブジェクトを移動させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

20

【請求項 39】

上記ユーザ入力が入力されたオブジェクトを、上記インターフェースエレメントが入ることを防止されるゾーンである上記コンピューティングインターフェースの禁止されたゾーンの中に移動させるとき、上記インターフェースエレメントに関連する上記影響の領域の内部に位置する上記表示されるオブジェクトを、上記インターフェースエレメントから分離させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 40】

30

上記ユーザ入力が入力されたインターフェースエレメントのアトラクション機能を起動させるとき、上記インターフェースエレメントに関連する上記影響の領域内のすべての表示されるオブジェクトを、上記影響の領域の中心に向かって移動させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 41】

上記ユーザ入力が入力されたインターフェースエレメントのアトラクション機能を起動させるとき、上記コンピューティングインターフェース内のすべての表示されるオブジェクトを、上記インターフェースエレメントに関連する上記影響の領域の中心に向かって移動させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 42】

40

上記ユーザ入力が入力されたインターフェースエレメントのアトラクション機能を起動させるとき、選択された属性を有する表示されるオブジェクトのサブセットを上記インターフェースエレメントに関連する上記影響の領域の中心に向かって移動させることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 43】

上記インターフェースエレメントに関連する上記影響の領域の中心に到達する表示されるオブジェクトを隠すことを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 44】

上記受信された入力が入力されたインターフェースエレメントの動作を停止させるとき、上記

50

インターフェースエレメントに関連する上記影響の領域の内部のすべての表示されるオブジェクトを上記影響の領域の中心から遠ざけることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 45】

上記受信された入力が入力インターフェースエレメントの反発機能を起動させるとき、上記コンピューティングインターフェース内のすべての表示されるオブジェクトを上記インターフェースエレメントに関連する上記影響の領域の中心から遠ざけることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 46】

上記受信された入力が入力インターフェースエレメントの反発機能を起動させるとき、選択された属性を有する表示されるオブジェクトのサブセットを上記インターフェースエレメントに関連する上記影響の領域の中心から遠ざけることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 47】

上記受信された入力が入力インターフェースエレメントの削除機能を起動させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の表示されるオブジェクトを削除することを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 48】

上記受信された入力が入力インターフェースエレメントをサイズ変更させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の表示されるオブジェクトをサイズ変更することを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【請求項 49】

上記受信された入力が入力インターフェースエレメントの動作を停止させるとき、上記インターフェースエレメントに関連する上記影響の領域内部の隠された表示されるオブジェクトを見せることを含む請求項 27 から 36 のうちのいずれか 1 つの請求項記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、コンピューティングシステムのためのインターフェースに関し、特に、コンピュータインターフェース上のオブジェクトを操作するためのシステム、方法、及びインターフェースに関するが、これに限らない。

【背景技術】

【0002】

対話型のデスクトップインターフェースは、現在のコンピュータのオペレーティングシステムの一部として知られている。いくつかのコンピューティングアプリケーションでは、対話型のテーブルトップインターフェースを有することが望ましい。対話型のテーブルトップインターフェースは、テーブルトップディスプレイ上で 1 人又はそれ以上のユーザとの相互作用を可能にする。対話型のテーブルトップインターフェースは、多数のユーザによるデジタル写真などのオブジェクトの協調的な共有を促進する。このようなインターフェースは、テーブルトップディスプレイ上に投影される画像を備える。例えば、画像は、埋め込みスクリーン又はデータプロジェクタの使用によって投影されてもよい。1 人又はそれ以上のユーザは、例えば、表示される画像のオブジェクトを操作するために、インターフェースの入力を介して表示される画像と相互作用してもよい。例えば、インターフェースへの入力は、画像が投影されるテーブルトップの接触感知面 (touch sensitive surface) によって提供される。この形式のコンピュータインターフェースは、いわゆる「広範囲に普及した (pervasive)」コンピューティングを促進する。

【先行技術文献】

【非特許文献】

10

20

30

40

50

【 0 0 0 3 】

【非特許文献 1】Margaret R. Minsky, "Manipulating simulated objects with real-world gestures using a force and position sensitive screen", SIGGRAPH Computer Graphics, 18 (3), pp. 195-203, 1984. ISSN0097-8930. doi: <http://doi.acm.org/10.1145/800031.808598>

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 4 】

しかしながら、多くのオブジェクトがテーブルトップコンピュータのインターフェースに表示されるとき、インターフェースは乱雑となる。これは、ユーザが表示されるオブジェクトと容易に相互作用することを妨げ、かつユーザの生産性を減少させる。

【課題を解決するための手段】

【 0 0 0 5 】

本発明の第 1 の態様によれば、コンピューティングインターフェースのインターフェースエレメントとの相互作用が可能な表示されるオブジェクトを操作する方法であって、上記コンピューティングインターフェースは、上記表示されるオブジェクトを表示エリアに表示するディスプレイモジュールと、ユーザ入力を受信する入力モジュールとを有し、上記方法は、上記表示されるオブジェクトを上記インターフェースエレメントと相互作用させるユーザ入力を受信することと、上記表示されるオブジェクトを上記受信された入力の性質にしたがって操作することとを含み、上記操作は、上記インターフェースエレメントに対して上記表示されるオブジェクトのサイズを変化させることを含む方法を提供する。

【 0 0 0 6 】

一実施形態では、上記受信された入力が、上記表示されるオブジェクトの上記インターフェースエレメントに属するとして指定された上記表示エリアの領域の中への移動、領域の外への移動、又は領域を通った移動となるとき、上記表示されるオブジェクトのサイズが変化する。

【 0 0 0 7 】

一実施形態では、上記ユーザ入力は、上記ユーザ入力が上記表示されるオブジェクトを直接的に移動させる相互作用となる方法で、上記表示エリア内での上記表示されるオブジェクトの移動となる。

【 0 0 0 8 】

一実施形態では、上記ユーザ入力は、上記表示されるオブジェクトを上記インターフェースエレメントの上記領域の中へ、上記領域の外へ、又は上記領域を通して移動させる。

【 0 0 0 9 】

一実施形態では、上記ユーザ入力は、間接的に上記表示されるオブジェクトの移動となる。

【 0 0 1 0 】

一実施形態では、上記ユーザ入力は、上記インターフェースエレメントの上記領域を上記表示エリア内で再定義し、上記インターフェースエレメントと上記表示されるオブジェクトとの相互作用となる。

【 0 0 1 1 】

一実施形態では、上記インターフェースエレメントの上記領域を再定義することは、上記領域を上記表示エリア内で移動する形式である。

【 0 0 1 2 】

一実施形態では、上記インターフェースエレメントの上記領域を再定義することは、上記領域を上記表示エリア内でサイズ変更する形式である。

【 0 0 1 3 】

一実施形態では、上記ユーザ入力は、上記インターフェースエレメントの操作属性 (manipulation property) の起動を示し、上記インターフェースエレメントと上記表示されるオブジェクトとの相互作用となる。

【 0 0 1 4 】

一実施形態では、上記オブジェクトのサイズは、上記領域を通る移動の方向にしたがって変化させられる。好ましくは、上記方向は、半径方向である。

【 0 0 1 5 】

一実施形態では、上記オブジェクトのサイズは、上記オブジェクトが上記領域の中心に向かって移動されるにつれて縮小される。一実施形態では、上記オブジェクトのサイズは、上記オブジェクトが上記領域の中心から遠ざけられるにつれて拡大される。

【 0 0 1 6 】

一実施形態では、上記方法は、複数の表示されるオブジェクトを上記受信された入力の性質にしたがって操作することを含む。

10

【 0 0 1 7 】

一実施形態では、影響はさらに、上記領域の中へ、上記領域の外へ、又は上記領域を通じて移動するすべての表示されるオブジェクトのサイズを変化させることを含む。

【 0 0 1 8 】

一実施形態では、影響はさらに、上記領域が移動されるときに、上記領域の内部に位置するすべての表示されるオブジェクトを上記領域とともに移動させることを含む。

【 0 0 1 9 】

一実施形態では、影響はさらに、上記領域が移動されるときに、上記領域が上記表示エリアの禁止されたゾーンに遭遇しない場合、上記領域の内部に位置するすべての表示されるオブジェクトを上記領域とともに移動させることを含む、上記領域が上記表示エリアの禁止されたゾーンに遭遇した場合、上記領域は、上記禁止されたゾーンに入ることを防止され、かつ上記表示されるオブジェクトは、上記受信された入力にしたがって移動し続ける。

20

【 0 0 2 0 】

一実施形態では、上記操作はさらに、上記ユーザ入力が上記インターフェースエレメントの操作属性の起動を示すとき、上記領域内のすべての表示されるオブジェクトを上記領域の中心に向かって移動させることを含む。

【 0 0 2 1 】

一実施形態では、上記操作はさらに、上記ユーザ入力が上記インターフェースエレメントの操作属性の起動を示すとき、上記表示エリア内のすべての表示されるオブジェクトを上記領域の中心に向かって移動させることを含む。

30

【 0 0 2 2 】

一実施形態では、上記操作はさらに、上記ユーザ入力が上記インターフェースエレメントの操作属性の起動を示すとき、選択された属性 (property) を有する表示されるオブジェクトを上記領域の中心に向かって移動させることを含む。

【 0 0 2 3 】

一実施形態では、上記操作はさらに、上記領域の中心に到達する表示されるオブジェクトを隠すことを含む。

【 0 0 2 4 】

一実施形態では、上記操作は、上記受信された入力が上記インターフェースエレメントの操作属性の起動を示すとき、上記領域内のすべての表示されるオブジェクトを上記領域の中心から遠ざけることである。

40

【 0 0 2 5 】

一実施形態では、上記操作は、上記受信された入力が上記インターフェースエレメントの操作属性の起動を示すとき、上記表示エリア内のすべての表示されるオブジェクトを上記領域の中心から遠ざけることである。

【 0 0 2 6 】

一実施形態では、上記操作はさらに、上記受信された入力が上記インターフェースエレメントの操作属性の起動を示すとき、選択された属性を有する表示されるオブジェクトを上記領域の中心から遠ざけることを含む。

50

【 0 0 2 7 】

－実施形態では、上記操作はさらに、上記受信された入力が入記インターフェースエレメントの操作属性の起動を示すとき、上記領域内部の表示されるオブジェクトを削除することを含む。

【 0 0 2 8 】

－実施形態では、影響はさらに、上記受信された入力が入記インターフェースエレメントのサイズ変更を示すとき、上記領域内部の表示されるオブジェクトをサイズ変更することを含む。好ましくは、上記オブジェクトの上記サイズ変更は、上記インターフェースエレメントの上記サイズ変更と同一のタイプ及び／又は比例するものである。－実施形態では、上記オブジェクトの上記サイズ変更は、上記インターフェースエレメントの上記領域のサイズに比例する。

10

【 0 0 2 9 】

－実施形態では、上記操作はさらに、上記受信された入力が入記インターフェースエレメントの操作属性の起動を示すとき、上記領域内部の隠された表示されるオブジェクトを見せることを含む。

【 0 0 3 0 】

－実施形態では、上記インターフェースエレメントの表現は、上記領域内部に表示される。－実施形態では、表現はブラックホールである。

【 0 0 3 1 】

本発明の第2の態様によれば、コンピューティングインターフェースのインターフェースエレメントと相互作用する表示エリア内に表示されるオブジェクトを操作するコンピュータプログラムを記憶するコンピュータ読み取り可能な記憶媒体であって、上記コンピュータプログラムは、コンピュータによって実行されるときに、上記コンピュータに、上記表示されるオブジェクトを上記インターフェースエレメントと相互作用させるユーザ入力を受信させ、上記表示されるオブジェクトを上記受信された入力の性質にしたがって操作させる命令を備え、上記操作は、上記受信された入力が入記表示されるオブジェクトの上記インターフェースエレメントに属するとして指定された上記表示エリアの領域の中への移動、領域の外への移動、又は領域を通った移動となるとき、上記表示されるオブジェクトのサイズを変化させることを含むコンピュータ読み取り可能な記憶媒体を提供する。

20

【 0 0 3 2 】

本発明の第3の態様によれば、コンピューティングインターフェースのインターフェースエレメントと相互作用する表示されるオブジェクトを操作する装置であって、上記装置は、表示エリアを有するディスプレイと、ユーザ入力と、プロセッサとを備え、上記プロセッサは、上記表示されるオブジェクトを上記インターフェースエレメントと相互作用させるユーザ入力を受信し、上記表示されるオブジェクトに上記受信された入力の性質にしたがって影響を及ぼすように構成され、上記操作は、上記受信された入力が入記表示されるオブジェクトの上記インターフェースエレメントに属するとして指定された上記表示エリアの領域の中への移動、領域の外への移動、又は領域を通った移動となるとき、上記表示されるオブジェクトのサイズを変化させることを含む装置を提供する。

30

【 0 0 3 3 】

本発明の第4の態様によれば、コンピューティングインターフェースのインターフェースエレメントと相互作用する表示されるオブジェクトを操作する装置であって、上記装置は、表示エリアを有するディスプレイと、ユーザ入力と、上記表示されるオブジェクトを上記インターフェースエレメントと相互作用させるユーザ入力を受信する手段と、上記表示されるオブジェクトを上記受信された入力の性質にしたがって操作する手段とを備え、上記操作は、上記受信された入力が入記表示されるオブジェクトの上記インターフェースエレメントに属するとして指定された上記表示エリアの領域の中への移動、領域の外への移動、又は領域を通った移動となるとき、上記表示されるオブジェクトのサイズを変化させることを含む装置を提供する。

40

【 0 0 3 4 】

50

本発明の第５の態様によれば、表示エリアをカバーする表示を生成するディスプレイモジュールとユーザ入力を受信する入力モジュールとを有するコンピューティングインターフェースのためのインターフェースエレメントであって、上記エレメントは、上記表示されるオブジェクトを上記インターフェースエレメントと相互作用させるあるユーザ入力をインターフェースエレメント入力として指定するインターフェースエレメント相互作用モジュールと、インターフェースエレメント入力を受信されたとき、上記表示内のオブジェクトを上記受信された入力の性質にしたがって操作するオブジェクト操作モジュールとを備え、上記操作モジュールは、表示されるオブジェクトを上記領域の中へ、上記領域の外へ、又は上記領域を通して移動させるにつれて、上記入力が見す表示されるオブジェクトのサイズを変化させるように構成されるインターフェースエレメントを提供する。

10

【００３５】

本発明のより良い理解を提供するために、好ましい実施形態が、添付の図面を参照して単なる例として説明されるであろう。

【図面の簡単な説明】

【００３６】

【図１】本発明の実施形態に係るインターフェースエレメントを提供するように設けられたコンピューティングシステムの概略図である。

【図２】本発明の実施形態に係るインターフェースエレメントを表示するインターフェースのスクリーンショットである。

【発明を実施するための形態】

20

【００３７】

本発明の一態様は、コンピュータインターフェースのスクリーンの乱雑さを削減する新しいインターフェースエレメントを提供する。本明細書で説明される実施形態では、コンピュータインターフェースは、テーブルトップコンピューティングシステムによって提供される表面実装（surface-mount）ディスプレイに表示される。インターフェースエレメントは、ユーザ入力に基づいて、インターフェース上に表示されるオブジェクトを操作するように、又はオブジェクトに影響を与えるように動作する。一実施形態では、オブジェクトを操作することは、受信された入力が、オブジェクトがインターフェースエレメントに関連したディスプレイの領域の中へ、領域の外へ、又は領域を通して移動される結果となると、表示されるオブジェクトのサイズを変化させることを含む。例えば、テーブルトップコンピュータのユーザが、ユーザインターフェース上に現在表示されている１０個のオブジェクト（例えば、デジタル写真）のうちの１つに焦点を当てることを望む場合、ユーザは、他の９個のオブジェクトのうちの任意の１つ又はそれ以上をインターフェースエレメントに向けてドラッグして、それらのサイズを効果的に最小化してもよい。

30

【００３８】

上述した機能性を実行するために、図１を参照して、テーブルトップコンピュータ１０２は、マザーボード１１０、中央処理装置１１２、ランダムアクセスメモリ１１４、ハードディスク１１６、及びネットワークハードウェア１１８を含むコンピュータハードウェアを備える。テーブルトップコンピュータ１０２はまた、画像（すなわち、ユーザインターフェース）をテーブルトップの表面上に投影するプロジェクタの形式でディスプレイ１０９を含む。テーブルトップコンピュータ１０２はまた、ユーザ入力を入力装置から受信する入力モジュール１１１を含む。入力装置を用いて、テーブルトップコンピュータのユーザは、ユーザインターフェース上に表示されるオブジェクトを操作する。

40

【００３９】

ハードウェアに加えて、テーブルトップコンピュータ１０２は、ハードディスク上に存在し、かつハードウェアと協働して、ソフトウェアアプリケーションが実行される環境を提供する（マイクロソフトコーポレーションによって製造されかつライセンスを与えられているマイクロソフトウィンドウズ（登録商標）ＸＰオペレーティングシステムなどの）オペレーティングシステムを含む。この点で、テーブルトップコンピュータ１０２のハードディスク１１６は、ユーザインターフェースを表示するビデオ出力装置を制御するディ

50

スプレイモジュールを有する。

【0040】

本明細書で説明される実施形態では、オブジェクトはデジタル写真の形式であるが、テキストファイル又はコンピュータで生成されたグラフィック画像などの他のオブジェクトが、等しく表示されかつ操作されることが理解されるであろう。一実施形態では、インターフェース111への入力、画像が投影されるテーブルトップの接触感知面によって提供される。

【0041】

本明細書で説明される実施形態では、オブジェクトは、テーブルトップ上に表示可能なデジタル画像の形式である。画像は、口語的に「写真」と呼ばれてもよい。写真は、長方形のオブジェクトとして表示エリア内に表示される。写真は、指、ペン、スタイラス、又はカーソルなどのポインタを用いて「ドラッグ」の動作を模倣することによって、テーブルトップの全域で移動されることができる。ポインタの接触は、入力装置によって受信され、かつ入力として解釈される。写真は、入力に依存して、例えば、移動、回転、及びサイズ変更などの他の方法で操作されることができる。

【0042】

多くの写真が表示される場合、表示エリアが乱雑になる。これは、ユーザの生産性を減少させる。したがって、実施形態は、現在ユーザによって必要とされていないオブジェクトを「保持する」ことができるインターフェースエレメントを提供する。保持されたオブジェクトは、サイズ変更されて、乱雑さの削減を支援する。さらに、本発明の一実施形態は、インターフェースエレメントが写真を表示エリアから一時的に除去することを可能にする。別の実施形態では、写真は永久に削除される。

【0043】

図2を参照すると、絵を用いて表されたインターフェースエレメント212を表示するテーブルトップコンピュータのユーザインターフェース210のスクリーンショットが表示されており、今後、ブラックホールと呼ばれる。ブラックホールは、典型的に円形であり、そのような形状はユーザに即座に認識可能である。しかしながら、ブラックホールは、任意の適切な形式をとってもよいことが理解されるであろう。ブラックホール212は、インターフェースの一部を形成する。ブラックホール212は、ユーザインターフェース210の一部である影響の領域を有する。影響の領域に入る、又は領域内に移動する（写真などの）オブジェクト214は、ブラックホール212の影響特性によって影響を受ける。ブラックホール212は、影響の領域を分割する中心部分218及び周辺220を備える。一実施形態では、オブジェクトが中心部分の内部である（又は中心部分に接触している）か、又は（中心部分に接触することなく）周辺の内部である（又は周辺に接触している）かに依存して、領域内のオブジェクトに対して、中心部分はあるタイプの影響を有し、かつ周辺は別のタイプの影響を有する。

【0044】

周辺220の内部で移動するオブジェクト214は、ユーザ入力オブジェクトをブラックホールの中心に向かって移動させるか、又はブラックホールの中心から遠ざけるかにしたがって、自動的にそのサイズが変更される。サイズ変更は、オブジェクトを移動させるユーザ入力への実時間のフィードバックを与えるために、即座であり、かつ移動に比例する。特に、オブジェクトが中心に向かって移動するとき、オブジェクトはサイズが縮小され、かつオブジェクトが中心から遠ざかるとき、オブジェクトは拡大される。この連続的なフィードバックは、移動の方向に依存して、写真がブラックホールに「吸い込まれる」、又はブラックホールから「吐き出される」という外観を与える。

【0045】

一実施形態では、中心部分は、その内部に移動されたオブジェクトを隠す。もう1つの実施形態では、中心部分に接触するオブジェクトは、ディスプレイから削除される。中心部分は、ブラックホールの幾何学的な中心点であってもよく、又は（小さい）中心のエリアであってもよい。

10

20

30

40

50

【 0 0 4 6 】

図 2 は、ブラックホール 2 1 2 の使用を概略的に示し、図 2 では、写真 2 1 4 が指型のカーソルポインタ 2 1 6 によって示されるように選択され、かつブラックホール 2 1 2 の中心に向かって移動される。ブラックホールは、オブジェクトがホールの中心に向かって移動するにつれて、写真のサイズを縮小することによって、オブジェクトの表示されるサイズに影響する。これは、現在使用されていない写真が邪魔にならないように移動され、又は削除されることができるので、テーブルトップインターフェース 2 1 0 の乱雑さを制御することを支援する。

【 0 0 4 7 】

インターフェースは、ディスプレイ内でのオブジェクトの位置、スケール、及び方向をメモリに維持することによって、オブジェクトを表示する。ディスプレイは、OpenGLなどのグラフィックAPIを用いて繰り返し再描画される。ディスプレイが再描画されるごとに、OpenGLは、正しい位置並びに方向で、そのオブジェクトのためのテクスチャ、及び装飾を描画する。相互作用が発生したとき、インターフェースは、どのオブジェクトが選択されたか、及びオブジェクトのどこで選択が発生したかを決定する。

10

【 0 0 4 8 】

インターフェースは、その後、入力を選択されたオブジェクトをどのように操作するかを決定する。これは、入力を選択されたオブジェクトを移動させるか、及びどのように移動させるかを決定することを含む。インターフェースエレメントは、オブジェクトの移動が、オブジェクトをブラックホールによって影響を及ぼされるようにするかを決定する。これを実装するための(C++コードの形式の)命令の例として付録Aを参照する。代替として、ブラックホールが選択された場合、インターフェースエレメントは、ブラックホールの操作が他のオブジェクトに影響するかを決定する。

20

【 0 0 4 9 】

図 2 では、ユーザがオブジェクト 2 1 4 の上方のテーブルトップ又はスクリーンに接触し(又はマウス、トラックボール、又は他の入力装置によって指図されるカーソルを用いて接触し)、かつオブジェクトをブラックホールの影響の領域の中へドラッグすることが示されている。オブジェクトの中心がブラックホールの中心に接近するよりは、指ポインタ 2 1 6 がブラックホールの中心に接近する。オブジェクトの中心は、ちょうどポインタの移動とともに動く。ブラックホールは、入力オブジェクトをこの影響の領域の内部に移動させ、かつオブジェクトが移動するにつれて、オブジェクトをサイズにおいて縮小させることを認識する。ポインタが中心部分に到達すると、写真は隠される。例えば、以下の付録Bを参照。

30

【 0 0 5 0 】

ブラックホールの周辺に置かれた写真は、単純にそれらを選択し、かつ外にドラッグすることによって回復される。これは、写真がブラックホールによって影響される前のサイズに戻るように写真を拡大するであろう。

【 0 0 5 1 】

ブラックホールがサイズ変更された場合、影響の領域内部の写真は、そのサイズを反比例して変更する。一実施形態では、これは、影響の領域内のオブジェクトのスケールされたサイズがブラックホールのサイズに比例して縮小されることに起因し、したがって、ブラックホールのサイズが小さくなるにつれて、オブジェクト上のその縮小効果が減少する。したがって、ブラックホールから写真を除去するもう1つの方法は、ブラックホールのサイズを縮小することであり、その結果、ブラックホール内部の写真は、サイズが拡大し、かつ選択され、ブラックホールの影響の領域から外にドラッグされる。

40

【 0 0 5 2 】

ブラックホールの影響の領域に置かれたオブジェクトは、取り付けられる。言い換えると、ブラックホールが移動される場合、オブジェクトはブラックホールとともに移動する。

【 0 0 5 3 】

50

アルゴリズムが使用されて、ブラックホールの周りの写真の階層化を決定する。あるアルゴリズムは、ブラックホールが移動されるときに、ブラックホール内のオブジェクトを保ち、別のアルゴリズムは、画像のサイズ、ブラックホールのサイズ、及びそれぞれの画像の中心とブラックホールの中心との間のユークリッド距離に基づいて、影響の領域内部のオブジェクトのサイズを決定する。

【 0 0 5 4 】

オブジェクトをブラックホールから除去するもう1つの方法は、ユーザが、パーソナルスペースなどのブラックホールが入ることを禁止されている表示エリアのゾーンに、ブラックホールを移動させることを試みることである。ブラックホールは、パーソナルスペースに移動されることができないが、ブラックホールに張り付いたオブジェクトは、そのスペースに入ることができ、かつユーザがペン/スタイラスなどを移動させ続けるにつれて、外にドラッグされる。したがって、ブラックホールは、パーソナルスペースの境界に引っ掛かり、一方、写真は、パーソナルスペースの中へ移動される。

10

【 0 0 5 5 】

ユーザがブラックホールの選択された位置を持つとき、これは、ブラックホールの1つ又はそれ以上の操作を起動させることができる。一実施形態では、操作特性は、影響の領域内のすべてのオブジェクトを永久に削除する。別の実施形態では、操作特性はまた、テーブルトップ上の他の画像をブラックホールに向かって連続的に移動させてもよく、「重力」の外観を提供する。そのような他の操作特性は、ブラックホール内部の画像を放射状に移動させ、かつテーブルトップ上の他の画像をブラックホールから遠ざけることであり、「反重力」の外観を提供する。他のものは、重力が特定の特性を有するオブジェクトにのみ影響することである。

20

【 0 0 5 6 】

様々な他の操作特性が、当業者には容易に明らかとなるであろう。このような操作特性は、ユーザに直観的なフィードバックを提供する。

【 0 0 5 7 】

いくつかの実施形態では、連続的な移動は一定の速度で発生する。もう1つの実施形態では、入力がアクティブな間、連続的な移動は加速する。オブジェクトはその後、「はじくこと (flicking)」に関係して後述するように、「仮想の運動量」を与えられてもよく、かつ「仮想の摩擦」にしたがってもよい。一実施形態では、連続的な移動は、ブラックホールの中心の領域の接線に沿う。一実施形態では、それぞれのオブジェクトの移動の速度は、ブラックホールの中心点とオブジェクトの中心点との距離の逆数に比例する。

30

【 0 0 5 8 】

テーブルトップ上のオブジェクトは、「はじくこと」にしたがってもよい。はじくことは、オブジェクトに関連した運動量を提供し、運動量は、仮想の摩擦力によって反対に作用される。(はじくことは、非特許文献1で説明されており、参照により本明細書に含まれる。)

【 0 0 5 9 】

画像がブラックホールの方向へはじかれ、かつ画像が影響の領域に遭遇するとき、仮想の重力がシミュレーションされ、オブジェクトをブラックホールの中心部分に向かって加速させる。使用されるアルゴリズムは、オブジェクトの運動量を維持し、かつブラックホールの中心へ向かう加速度を適用する。アルゴリズムは、重力の物理学を厳密にモデリングすることができ、ユーザの行動への現実的なフィードバックを維持する。したがって、オブジェクトは簡単に消滅せず、むしろ、オブジェクトがブラックホールに衝突するとき、オブジェクトは、ブラックホールの中心に向かって内側にらせん運動をするであろう。

40

【 0 0 6 0 】

オブジェクトが、進行中の軌道に留まることが可能である。しかしながら、アルゴリズムは含まれた項を有することができ、その結果、成分方向の速度における変化が速度と逆方向であるとき、中心へ向かう加速度は増大する。これは、オブジェクトが直接的に中心へ向かうことを熱心に目標としない場合でも、オブジェクトの経路に、ブラックホールの

50

中心に向かってらせん運動をすることを助力させる。オブジェクトが中心部分に遭遇するとき、オブジェクトは捕捉されかつ隠される。したがって、画像は一般に、ブラックホールに向かってはじかれることができ、目標が完全に正確でない場合でもブラックホールに吸い込まれるであろう。言い換えると、オブジェクトは、影響の領域に遭遇する限り、中心の領域に向けられる必要はなく、ブラックホールに吸い込まれるであろう。これは、ブラックホールが手の届くところでない場合でさえ、ユーザがオブジェクトを削除することを可能にする。例えば、付録 C 及び D を参照。

【 0 0 6 1 】

オブジェクトのはじきが強すぎて、かつオブジェクトが十分な運動量を有する場合、オブジェクトは、ブラックホールに吸い込まれることなく、ブラックホールを通過するであろう。これは、重力スリングショット効果 (gravitational slingshot effect) となる。しかしながら、別の変更例は、アルゴリズムにスリングショットを防止させるようにすることができる。

10

【 0 0 6 2 】

一実施形態では、ブラックホールは、はじかれることを防止される。すなわち、オブジェクトは、テーブルトップの全域で移動されることができ、ブラックホールは、運動量を得ないであろう。ブラックホールが解放されると、それは単に解放された場所に停止するであろう。しかしながら、ブラックホール内部のオブジェクトは、運動量を維持することができる。したがって、ブラックホールが停止したとき、その内部のオブジェクトは、それらの運動量の効力によって移動し続けることができ、したがって、ブラックホールから落ちる。代替として、ブラックホール内部のオブジェクトは、静止慣性 (stationary inertia) を維持することができ、したがって、ブラックホールが非常に素早く移動させられるとき、オブジェクトは、緩やかに移動する傾向があり、ブラックホールから落ちるであろう。

20

【 0 0 6 3 】

(ブラックホールオブジェクトを含む)それぞれのスクリーン上のオブジェクトについて、以下が維持される。

(x, y, z) 「世界」座標でのオブジェクトの中心の位置。(z は、描画されてもよいオブジェクトの順序を決定する。)

(x, y) オブジェクトが移動されている場合、「オブジェクト」座標での接触点の位置。

30

「オブジェクト」座標は、それぞれのオブジェクト上で [- アスペクト、 - 0 . 5] から [アスペクト、 0 . 5] まで広がる。

s , オブジェクトの所望のスケールであり、オブジェクト上の回転 / サイズ変更の操作を実行することによって変更可能。

$userLock$, オブジェクトを移動しているユーザの識別子、又は「NO__USER」。

【 0 0 6 4 】

ブラックホール以外のオブジェクトについて、以下のものがそれぞれのスクリーンの再描画で計算される。

40

bhd , 最近に決定されたブラックホールからの「距離」であり、下式のように計算される。

【 0 0 6 5 】

【 数 1 】

$$bhd = \frac{d\sqrt{2}}{s}$$

【 0 0 6 6 】

ここで、 d は、ブラックホールの中心とオブジェクトとの間のユークリッド距離の 2 乗であり、 s は、ブラックホールの現在のスケールである。

50

$bhd < 1.0$ のとき、オブジェクトは、ブラックホールの中心の部分にあるといわれる。

ブラックホールの周辺は、ブラックホール周辺の円形のエリアであり、 $bhd < 1.0$ である。

【0067】

ブラックホール以外のオブジェクトについて、以下の計算が実行される。

$residual$, オブジェクトがブラックホールの内部に移動され、続いてブラックホールの周辺で解放されたかの指示であり、それが発生したときの bhd の値 ($residual_bhd$) 。

オブジェクトが移動以外によってブラックホールの周辺に入る場合 (又はブラックホール自体が移動されて、その周辺がオブジェクトの中心を包囲する場合)、その $residual_bhd$ は、 1.0 である。

【0068】

オブジェクトがブラックホールの中心の部分にあるとき、表示されるスケールは、(所望のスケールを処理した後、) 以下のように変更される。

表示されるスケールは、 $bhd \times residual_bhd$ の係数によって削減される。

さらに、オブジェクトが現在移動されている (すなわち、接触されている) 場合、スケール操作の「中心」は、オブジェクトの中心 (すなわち、オブジェクト座標における点 $(0, 0)$) ではなく、代わりに接触点であり、その結果、オブジェクトが再スケールされるとき、接触点のオブジェクト座標は、変化しない。

【0069】

オブジェクトを表示するために、世界座標からスクリーン座標への変換が存在する。

【0070】

階層化の2つの概念が使用され、オブジェクトの最終的な描画順序を決定する - z 座標及びレイヤ。最後に描画されるオブジェクトは、オブジェクトの部分が部分的に又は全体的に透明でない場合、その下の任意のオブジェクトを塞いで示されるであろう。例えば、ブラックホールは、その縁の周りが透明であり、中心が不透明であり、それらの間の透明性の傾斜を伴っている。オブジェクトの z 座標は、最終的な描画順序 (オブジェクトは、 z 座標が減少する順序で描画される。) を決定する。

【0071】

オブジェクトが選択されるといつでも、そのオブジェクトは、同一のレイヤにおける任意のオブジェクトの最小の z 座標を与えられ、したがって、そのオブジェクトは、それらのオブジェクトの上に現れる。ブラックホールなどの特別なオブジェクトは、より高いレイヤを与えられる。したがって、ブラックホールは常に、他のオブジェクトの上に表示され、かつ決して覆い隠されない。

【0072】

オブジェクトは、それらがテーブルに置かれた物理的な写真であるように、反転される (裏返される) ことができる。接触インターフェースでは、例えば、これは、2本の指をそれぞれ隣接する写真の角の画像上に置いて、その後、画像を横切って指をドラッグして、それを反転することによって達成される。「ホットスポット」(例えば、それぞれのオブジェクトの角に重ねられた三角形) を選択し、それを画像の他の部分にドラッグすることなどの代替の反転技術がまた、使用されることができる。

【0073】

オブジェクトを反転するための技術が付録 E から H に示される。これらは、引数として、例えばスタイラス又は指のロケーションである反転のために使用される現在の「制御点」であるスクリーン上の (x, y) ピクセル座標をとる。最初のステップは、これを、仮想環境において反転されるオブジェクトを位置付けるために使用されるキャッシュされた逆変換行列を用いるオブジェクト座標に変換することである (しかし、任意の反転の前に、回転変換が描画手続きにおいて実行される。)。 x 座標は正規化されて、 y 座標と同程

10

20

30

40

50

度にされる（すなわち、これらは等しい重みを与えられて、反転するための最初の方向を決定する。）。

【 0 0 7 4 】

反転がまだ開始されていない場合、方向（*x f l i p*）は、より大きい成分を有する方向（すなわち、画像の中心から最も遠い）として決定される。また、我々が水平反転のための中心の左、又は垂直反転のための中心の下に向いている場合、それは「逆の」反転である。反転するための距離（*d i s t*）は、（点が、最初に - 0 . 5 から 0 . 5 までに伸ばされるので）位置成分の 2 倍であり、したがって、*d i s t* は、[- 1 . 0 , 1 . 0] の範囲にある。反転するための角度は、*d i s t* の逆余弦である。我々は、状態を維持して、開始されると「同一の」方向への反転を継続し、かつ我々は、裏面に反転するか、又は「正面」に戻すかを決定する。画像の反転自体は、描画手続きで発生する。

10

【 0 0 7 5 】

反転されたオブジェクトがブラックホールである場合、ブラックホールの隠されたコンテンツが表示される。これは、ユーザがオブジェクトを選択し、ブラックホールの外にドラッグすることを可能にする。

【 0 0 7 6 】

上述した反転技術は、スマート・インターネット・テクノロジー・シーアールシー・プロプライエタリー・リミテッドの名称で出願された「コンピュータディスプレイ上のデジタル画像を操作するシステム及び方法」と題された関係するオーストラリア仮特許出願の主題であり、仮出願番号 A U 2 0 0 7 9 0 4 9 2 7 に一致し、参照により本明細書に含まれる。

20

【 0 0 7 7 】

例 .

以下の状況を考慮する。ユーザは、自分のデジタルカメラを本発明の実施形態に係るインターフェースを搭載したコンピュータに接続する。カメラは、最近の休日から撮影された多数の写真を保持しており、したがって、写真のためのサムネイルを含むオブジェクトがロードされる。サムネイルは、画像が議論することを所望されたものであるかを判断する、又は 2 つの密接に関係した写真のどちらかに決定するには、不十分な品質であることがある。オブジェクトのサムネイルをドラッグすることは、サムネイルのより高い品質のコピーを生成させる。しかしながら、ユーザは、彼らがこのコピーをもはや必要としないと決定し、したがって彼らはそれを削除することを望む。

30

【 0 0 7 8 】

従来のコンピューティングインターフェースでは、ユーザは、アイコンを「ごみ箱（*T r a s h C a n*）」又は「リサイクル箱（*Recycle Bin*）」にドラッグしてもよいが、仮想環境では、これは多くの問題を有する。「ごみ」は通常、ディスプレイ上の固定された位置に位置付けられる。物理的に大きな仮想インターフェース上では、ユーザは、オブジェクトのロケーションに到達することができないことがあり、したがって、彼らは、ごみの方向にオブジェクトをはじくことを望むことがある。代替として、彼らは、ごみオブジェクトを環境の周囲に移動させることを望むことがある。複数のユーザが存在してもよく、かつ「ごみ」が単一のユーザによってのみ方向付けられ、又は簡単にアクセス可能であることがある。特に、インターフェースが標準的なコンピュータの習慣に親しんでいないユーザに直観的であるように設計されている場合、ごみを開けて何かを検索することは、テーブルトップコンピューティングシステムでは非実用的である。削除を確認するための確認ダイアログ、又はアイテムをごみに移動させるためのコンテキストメニューを提供することはまた、非実用的である。ごみは、覆い隠されることがあり、かつ不慣れなユーザは、アイテムをごみの上に不注意に移動させ、それを偶然に削除することがある。オブジェクトは、「ごみ」オブジェクトよりも非常に大きくスクリーンに表示されてもよく、かつマウスカーソルなしで、何がごみに移動させられるべきかを示す方法が明らかでないことがある。ユーザはまた、画像を削除することなく、画像を部分的に隠すことを望んでもよい。

40

【 0 0 7 9 】

50

ブラックホールを用いて写真／画像（すなわち、オブジェクト）を削除するために、ユーザは、オブジェクトを（例えば、指又はスタイラスを用いて）ブラックホールに向かってドラッグする。ユーザの指（すなわち、画像の中心ではなく、画像が接触された場所に関わらず）が、ブラックホールの影響の内部（ $bhd < 1.0$ ）になると、オブジェクトは、サイズが縮小され始める。ユーザの指がブラックホールの中心に近づくにつれて、オブジェクトは、それがブラックホールの中心でもはや見えなくなるまで、より小さくなる。代替として、ユーザがサイズの縮小に気付き、かつオブジェクトを削除／隠すことを実際には望まないことを決定する場合、ユーザは即座にそれを戻すことができる。

【 0 0 8 0 】

ユーザはまた、ブラックホールを移動させることを望んでもよい。ユーザは、単にブラックホールをドラッグすることによってこれを実行する。ブラックホール及びそのすべてのコンテンツ（まだ可視であってもよいその周辺又はヘリのアイテムを含む）が、新しいロケーションに移動され、それらの相対的な位置を維持する。ブラックホールが移動するにつれて、ブラックホールの近くに来るオブジェクトは、その中心がヘリに入った場合、影響される。その後、ブラックホールが解放される場合、ブラックホールによって影響されたオブジェクトは、ブラックホールによって「捕捉」され、ブラックホールが仮想環境における電気掃除機と同種の方法で使用されることを可能にする。

【 0 0 8 1 】

ブラックホールの中に解放された後、オブジェクトは検索されることができる。このための技術は、画像が置かれたブラックホール「内への」距離に依存して、変化する。オブジェクトがブラックホールのヘリにある場合、オブジェクトのある部分がブラックホールの下から「突き出している」という条件で、オブジェクトは、外に移動されることがある。オブジェクトのどの部分も突き出していない場合、ブラックホールがより小さくされてもよく、オブジェクトへのその効果が減少する（すなわち、ブラックホールのスケールが減少されるにつれて、 bhd がより大きくなる。）。ヘリのオブジェクトがより大きくなると、その後、オブジェクトは遠ざけられることができる。しかしながら、ブラックホールがどれほど小さくされうるかについて制限が課されてもよい。オブジェクトがブラックホールの中心の非常に近くに存在する場合、オブジェクトは、ブラックホールが進むことができないパーソナルスペースなどの「投棄場（dumping ground）」に、ブラックホールを移動させることによって、検索されることができる。代替として、ブラックホールを繰り返しはじくことは、ブラックホール内部の画像を後に残させることができる。

【 0 0 8 2 】

要求されないが、図を参照して説明された実施形態は、開発者による使用のために、アプリケーションプログラミングインターフェース（API）を用いて、又は一連のライブラリとして実装されてもよく、かつ端末若しくはパーソナルコンピュータのオペレーティングシステム、又はポータブルコンピューティング装置のオペレーティングシステムなどの別のソフトウェアアプリケーションの内部に含まれてもよい。一般に、プログラムモジュールは、特定の機能を実行する又は実行を支援するルーチン、プログラム、オブジェクト、構成要素、及びデータファイルを含むので、ソフトウェアアプリケーションの機能性は、多数のルーチン、オブジェクト、及び構成要素にわたって分散されてもよく、実施形態と同一の機能性、及び本明細書で主張されるより広範な発明を達成することが理解されるであろう。このような変形例及び変更例は、当業者の範囲内である。

【 0 0 8 3 】

本発明の当業者には、多くの変更例が本発明の精神及び範囲を逸脱することなく実行されてもよいことが理解されるであろう。

【 0 0 8 4 】

[表 1]

付録 A

10

20

30

40

[language={C++}]

効果 1 - それぞれのオブジェクトがブラックホールによって影響されるかを決定するそれぞれのスクリーンの再描画での手続き。

```
bool Resource::blackHoled() {
    double dx = env->blackhole->position.x;
    double dy = env->blackhole->position.y;
    if (bh_moving
        || (worldMPos.x == 0.0 && worldMPos.y == 0.0)
        || userLock == NO_USER) {
        dx -= position.x;
        dy -= position.y;
    } else {
        dx -= worldMPos.x;
        dy -= worldMPos.y;
    }
    double old_bhd = blackhole_distsq;
    bool oldinBH = inBlackhole;
    blackhole_distsq = M_SQRT2*(dx*dx + dy*dy) / env->blackhole->scale;

    inBlackhole = blackhole_distsq < 1.0;
    wasInBlackhole = wasInBlackhole || inBlackhole;
    if ((inBlackhole && old_bhd != blackhole_distsq) ||
        inBlackhole != oldinBH) {
        pc.makeDirty(); //それがまだダーティでなければ、ブラックホールはおそらく
移動した。
    }

    if (residual) {
        //我々は、有効な b l a c k h o l e _ d i s t s q を変更することを実際には
望まない。
        residual_bh_distsq = old_bhd / blackhole_distsq;
        residual = false;
    }
    if (!inBlackhole) {
        //我々は、ブラックホールから取り出すとすぐに、これをリセットする。
        residual_bh_distsq = 1.0;
    }

    return
    (bh_trapped && blackhole_distsq < RConfig::MIN_SCALE_FOR_DRAW) ||
    (link && link->getParent()->isInBlackhole());
}
```

【 0 0 8 5 】

[表 2]

付録 B

ブラックホールに影響されたオブジェクトをどのようなサイズで表示するかを決定する手続き。

```

void Resource::wormHole() {
    inBlackhole = blackhole_distsq < 1.0;
    if (inBlackhole) {
        if (!bh_moving && (userLock != NO_USER || bh_trapped)) {
            //マウスカーソルの位置でスケールして、その相対的な位置を保つ。
            glTranslatef(currMPos.x, currMPos.y, 0);
        }
        glScalef(blackhole_distsq*residual_bh_distsq, // x スケール
                blackhole_distsq*residual_bh_distsq, // y スケール
                1.0); // z スケール
        if (!bh_moving && (userLock != NO_USER || bh_trapped)) {
            glTranslatef(-currMPos.x, -currMPos.y, 0);
        }
    }
    if (!inBlackhole && bh_trapped) {
        bh_trapped = false;
        userLock = NO_USER;
    }
}

```

【 0 0 8 6 】

[表 3]

付録 C

ブラックホール近傍での運動量のための位置更新手続き。

```

/**
 *
 * ¥ r e t u r n   t r u e   アニメーションが終了した場合。
 */
bool Momentum::rel_update(unsigned ms)
{
    //与えられた初速度及び加速度を経時的に変位させるための古典物理学の公式
    // ¥ f $ s = u t + ¥ f r a c { 1 } { 2 } a t ^ 2 ¥ f $ を使用する。

    float dt = 0.001*(ms - lastms);
    lastms = ms;

    if (r->selectedBy() != user) {
        //他の人がそれに接触すると、我々は停止する。
        if (r->selectedBy() >= 0)
            return true;
        //我々が非選択になると、我々は停止するときに、我々はまだ枠線色、
        //アクセス制限、及び非選択を必要とする。
        killselect = true;
    }
    //我々が同じユーザによって再び接触されたかを確認し、そうであれば、停止する。
    if (r->clickPos != clickPos)
        return true;

    //摩擦 / 抗力に起因する減速度は、__速度__の x / y 成分の反対方向に

```

```

//方向付けられる。大きさはちょうど d e c e l - - 摩擦 / 抗力に起因
//する ( 定数の ) 減速度である。
float vtheta = xv == 0.0 ? M_PI/2.0 : atanf(fabs(yv / xv));
float accel_x = (xv < 0 ? 1.0 : -1.0) * cosf(vtheta) * decel;
float accel_y = (yv < 0 ? 1.0 : -1.0) * sinf(vtheta) * decel;

//我々がブラックホールの近傍であるときに、大きさ
// B L A C K H O L E _ A C C E L のブラックホールの中心へ向かう成分を
//加算することによって、加速度ベクトルを変更する。
if (r->blackholeDist() < 1.0) {
    /*ブラックホールのワープの前に、我々はスクリーン位置を使用することに留意
    する。*/
    float dx = r->env->blackhole->getPC().getScreen().x
               - r->getPC().getScreen().x;
    float dy = r->env->blackhole->getPC().getScreen().y
               - r->getPC().getScreen().y;
    float theta = dx == 0.0 ? M_PI/2.0 : atanf(fabs(dy / dx));

    accel_x += (dx < 0 ? -1.0 : 1.0)
               * RConfig::BLACKHOLE_ACCEL
               * cosf(theta)
               * (dx * xv < 0.0 ? 1.5 : 1.0);

    accel_y += (dy < 0 ? 1.0 : -1.0)
               * RConfig::BLACKHOLE_ACCEL
               * sinf(theta)
               * (dy * yv > 0.0 ? 1.5 : 1.0);

}
//加速度ベクトルから速度及び変位を更新する。
float xvdiff = accel_x * dt;
float yvdiff = accel_y * dt;
float xdiff = xv * dt + 0.5 * accel_x * dt * dt;
float ydiff = yv * dt + 0.5 * accel_y * dt * dt;

xv = (fabs(xvdiff) >= fabs(xv) && r->blackholeDist() >= 1.0) ?
    0 :
    xv + xvdiff;

yv = (fabs(yvdiff) >= fabs(yv) && r->blackholeDist() >= 1.0) ?
    0 :
    yv + yvdiff;

if (!finite(xv) || !finite(yv)) {
    xv = yv = 0.0f;
}

// 10 ピクセル / 秒未満のときに停止する。 - - なぜ 10 か ? = > 我々がブラック
//ホールの中心によって「捕捉された」とき、フレームの再描画もまた停止する。
if (r->blackholeDist() < RConfig::BLACKHOLE_TRAPDIST ||

```

```

        (r->blackholeDist() >= 1.0 && fabs(xv) <= 20 && fabs(yv) <= 20)) {
    if (killselect)
        r->unSelect(user);
    if (r->blackholeDist() >= 1.0)
        r->settle();
    return true;
}
//我々の所望の位置を記憶する。
x0 = x0 + xdiff;
y0 = y0 + ydiff;

//その後、最も近いスクリーン / ピクセルのロケーションに移動し、境界に制限する
。
r->moveto(static_cast < int >(roundf(x0)),
          static_cast < int >(roundf(y0)));

if (r->getPC().getRealScreen().x + 3 >= r->env->getSurface()->w
    && RConfig::DATAWALL_SEND && !sent) {
    //トリガがスクリーンの右側で送信される。
    sent = true;
    datawall_send(r);
} else if (r->getPC().getRealScreen().x <= 3
    && RConfig::MAGICMIRROR_SEND && !sent) {
    //トリガがスクリーンの左側で送信される。
    sent = true;
    datawall_send(r, true);
}
return false;
}
【 0 0 8 7 】
[ 表 4 ]
付録 D

```

運動量アニメーションのトリガを制御する手続き。

```

void Mover::updatePositions() {
    if (positions.size() == RConfig::VELOCITY_WINDOW)
        positions.pop_back();
    positions.push_front(std::make_pair(current_time, current_xy_position));
}

MoveTracker* Mover::release() {
    if (!RConfig::DO_MOMENTUM
        || positions.size() < RConfig::VELOCITY_WINDOW
        || r->hasLink())
        return ResourceGesture::release();

    float dx = positions.front().second.x - positions.back().second.x;
    float dy = positions.front().second.y - positions.back().second.y;
    float dt = (positions.front().first - positions.back().first) / 1000.0f;

```

```

float vel_sq = (dx * dx + dy * dy) / (dt * dt);

if (vel_sq > RConfig::ESCAPE_VELOCITY && r != r->env->blackhole) {
    r->env->addAnimation(new Momentum(r,
                                     dx / dt, dy / dt,
                                     positions.front().second.x,
                                     positions.front().second.y));
}
return ResourceGesture::release();
}
【 0 0 8 8 】
[ 表 5 ]
付録 E

```

反転を開始するかを決定する手続き。

```

bool Image::flipSelected() {
    GLfloat x = fabsf(currMPos.x), y = fabsf(currMPos.y);
    return !inBlackhole &&
        (y - x > RConfig::CORNER_PROP ||
         x - y > 0.5*aspect - 0.5 + RConfig::CORNER_PROP);
}
【 0 0 8 9 】
[ 表 6 ]
付録 F

```

スクリーン上の現在の接触位置が与えられるとして、オブジェクトを反転する手続き。

```

bool Image::flipto(int screen_x, int screen_y) {
    P3<GLfloat> avgPos;
    //テクスチャにおいて、制御点が存在する座標を決定する。
    setObject(avgPos, screen_x, screen_y, true);
    avgPos.x /= aspect; // 0 から 1 までの間に正規化する。
    //(我々は常に、より大きな方向に沿って反転する。)

    float cosval, dist;
    if (!flipping) { //まだ反転していない。
        xflip = fabs(avgPos.y) > fabs(avgPos.x);
        dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
        backflip = dist < 0;
        wasflipped = flipped;
        flipping = true;
    } else {
        dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
    }

    //距離の符号が変化する場合、反転状態がトグルする。
    flipped = wasflipped != (backflip != dist < 0);

    //角度を決定して、画像を回転する。

```

```
cosval = fabs(dist);
if (cosval < 1.0) {
    flipangle = acosf(cosval); // 0 と 90 の間。

    //我々が「間違った」側で開始した場合、我々は反対に回転する。
    //反転された場合、我々はまた、中心で魔法の反転を実行する。
    if (xflip == backflip)
        flipangle *= -1.0;

    //我々が開始から変化を有するときはいつでも、我々はこちらにいて、
    //180度から減算することを望む。
    if (flipped != wasflipped)
        flipangle = -flipangle;

} else {
    //dist > 1.0 の場合、接触点は遠ざかった。 - - 反転しない。
    flipangle = 0.0;
}

pc.makeDirty();
return true;
}

void Image::drawPartialFlip() {
    glPushMatrix();
    glRotatef(flipangle*RAD2DEG,
              xflip ? 1.0 : 0.0,
              xflip ? 0.0 : 1.0,
              0.0);
    if (flipped) {
        drawReverse();
        //我々が反転している場合、背景も描画する。
        if (flipping) {
            glRotatef(180,
                      xflip ? 1.0 : 0.0,
                      xflip ? 0.0 : 1.0,
                      0.0);
            glCallList(*model);
        }

    } else {
        drawTexture();
        if (flipping) {
            glRotatef(180,
                      xflip ? 1.0 : 0.0,
                      xflip ? 0.0 : 1.0,
                      0.0);
            drawReverse(); /*レンダリングか??*/
        }
    }
}

glPopMatrix();
```

}
 【 0 0 9 0 】
 [表 7]
 付録 G

反転ウィジェットを使用してフレームのアスペクト比を変更する手続き。

```

bool SmartFrame::flipto(int x, int y) {
    GLint realy = flip(y);
    P3<GLfloat> avgPos;
    setWorld(avgPos, x, realy, true);

    //どちらのアスペクトが変更されるべきかを調査する。
    if (!flipping) {
        xflip = fabs(avgPos.y) > fabs(avgPos.x);
        wasflipped = flipped;
        flipping = true;
    }

    double dx = pc.getScreen().x - x;
    double dy = pc.getScreen().y - realy;
    if (fabs(dx) <= 1 && fabs(dy) <= 1)
        return false;

    double delta = sqrt(dx*dx + dy*dy);

    if (xflip) {
        // x f l i p が真であれば、中心のアスペクトは i n f であり、我々はスケール
        // を変更する必要がある。
        changeAspect(1.0 / (delta / (clickPosLine / (1.0 / clickAspect))));
        setScale(delta / (clickPosLine / clickScale), false, false);
    } else {
        changeAspect(delta / (clickPosLine / clickAspect));
    }

    pc.makeDirty();
    return true;
}
    【 0 0 9 1 】
    [ 表 8 ]
    付録 H
  
```

反転したオブジェクトの背景で、付着物の“フロー”レイアウトを実行する手続き。

```

void FlowLayout::doFlowLayout(Layout::COLL &col) {
    const double pad = padding * parent->getScale();
    const double mar = margin * parent->getScale() - pad; //負になることができる
    。

    const double maxw = parent->estWidth() - 2*mar;
  
```



```

const double maxh = parent->estHeight() - 2*mar;
double y = -(maxh * -0.5 + pad + mar);

COLL::iterator it = col.begin();
const COLL::iterator end = col.end();
const double rads = parent->getRads();

while (it != end) {
    COLL row;
    double h = (*it)->estHeight();
    double w = (*it)->estWidth() + pad;
    row.push_back(*it++); //少なくとも1つのエレメントを行に置く。
    for (; it != end; ++it) {
        //行を埋める。
        if (w + pad + (*it)->estWidth() > maxw)
            break;
        w += pad + (*it)->estWidth();
        //行の高さが増大されるべきかを調べる。
        if ((*it)->estHeight() > h)
            h = (*it)->estHeight();
        row.push_back(*it);
    }

    COLL::iterator rit = row.begin();
    const COLL::iterator rend = row.end();
    double x = maxw * -0.5 + pad + mar;
    y += -(h * 0.5);

    //ここで、オブジェクトの位置を調整する。
    for (; rit != rend; ++rit) {
        x += (*rit)->estWidth() * 0.5;
        if (!(*rit)->isManip()) {
            //操作されていないもののみを移動する。
            P3<GLfloat> oldPosition((*rit)->getPosition());
            GLfloat oldRot = (*rit)->getRads();

            (*rit)->rotaterad(rads);
            (*rit)->stealPosition().set(x*cos(-rads) - y*sin(-rads),
                                      x*sin(-rads) + y*cos(-rads),
                                      (parent->isFlipped() ?
                                      0.0001f :
                                      -0.0001f));
            (*rit)->stealPosition() += parent->getPosition();
            if (lastSettled == rit->addr()) {
                //我々が追加された最後のものであれば、移動をアニメーションさ
                せる。
                (*rit)->env->addAnimation(new LinearPath(*rit, oldPosition,
                (*rit)->getPosition(), 200), Animation::REPLACE);
                (*rit)->env->addAnimation(new LinearSpin(*rit, oldRot, rads,
                200), Animation::REPLACE);
            }
        }
    }
}

```

```

        lastSettled = 0;
    }
}
x += (*rit)->estWidth() * 0.5 + pad;
}
y += -(h * 0.5 + pad);
}
lastAdded = 0;
}

```

【図 1】

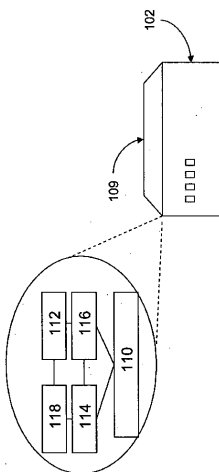


Fig. 1

【図 2】

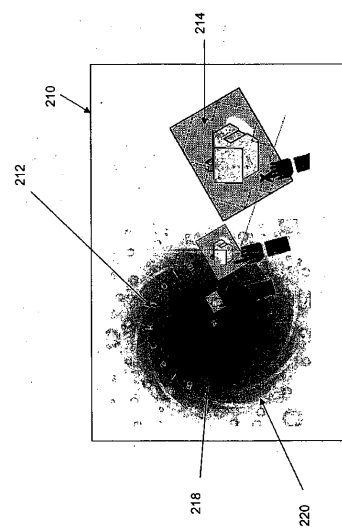


Fig. 2

フロントページの続き

(74)代理人 100125874

弁理士 川端 純市

(72)発明者 トレント・アプティド

オーストラリア 2 0 3 2 ニューサウスウェールズ州キングスフォード、シー・ストリート 6 / 1 -
3 番

審査官 篠原 功一

(56)参考文献 特開 2 0 0 2 - 1 4 0 1 4 7 (J P , A)

特開 2 0 0 4 - 2 5 8 8 3 8 (J P , A)

Hilliges, O.; Baur, D., Butz, A., Photohelix: Browsing, Sorting and Sharing Digital Photo Collections, Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop, IEEE, 2 0 0 7 年 1 0 月, p.87-p.94, URL, <http://research.microsoft.com/apps/pubs/default.aspx?id=132458>

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 3 / 0 0