

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5508269号  
(P5508269)

(45) 発行日 平成26年5月28日(2014.5.28)

(24) 登録日 平成26年3月28日(2014.3.28)

(51) Int.Cl.

F I

G 0 6 F 3/0481 (2013.01)

G 0 6 F 3/048 6 5 7 A

請求項の数 26 (全 16 頁)

(21) 出願番号 特願2010-524308 (P2010-524308)  
 (86) (22) 出願日 平成20年9月11日 (2008.9.11)  
 (65) 公表番号 特表2010-539568 (P2010-539568A)  
 (43) 公表日 平成22年12月16日 (2010.12.16)  
 (86) 国際出願番号 PCT/AU2008/001345  
 (87) 国際公開番号 W02009/033219  
 (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 2007904927  
 (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】 コンピュータディスプレイ上でデジタル画像を操作するためのシステム及び方法

## (57) 【特許請求の範囲】

## 【請求項 1】

デジタル画像を操作するためのユーザインタフェースシステムであって、  
 少なくとも一つのデジタル画像をディスプレイするように構成されたディスプレイを含み、

上記少なくとも一つのデジタル画像は、前側と、ユーザにディスプレイ可能な少なくとも一つの反対側を有し、前側と、反対側の各々は、少なくとも一つの検出部分を有し、

上記検出部分は、ユーザにより操作リクエストを検出し、該操作リクエストが検出されると、ディスプレイされている側が少なくとも部分的に回転して視界から消え少なくとも一つの反対側が少なくとも部分的にディスプレイされるルーティンを開始するように構成され、

回転の開始方向はリクエストに依拠して決定される、  
 ユーザインタフェースシステム。

## 【請求項 2】

上記検出部分は、上記前側と上記反対側の各々のうちの一つの規定された部分である  
 請求項 1 に記載のユーザインタフェースシステム。

## 【請求項 3】

操作リクエストがルーティンの実行の間にユーザに維持される場合のみ、完全な回転が維持される

請求項 1 又は 2 に記載のユーザインタフェースシステム。

## 【請求項 4】

ルーティンがルーティンの実行の間にユーザにフィードバックを与える  
請求項 1 乃至 3 のうちのいずれかに記載のユーザインタフェースシステム。

## 【請求項 5】

フィードバックが画像の回転のアニメーションである  
請求項 4 に記載のユーザインタフェースシステム。

## 【請求項 6】

アニメーションが、画像の表面に直交する軸に関して、画像の回転を描く  
請求項 5 に記載のユーザインタフェースシステム。

## 【請求項 7】

操作リクエストが、ユーザインタフェースシステムにインプットされる、手のジェスチャ、及び、スタイラス、ペン、若しくは電子周辺機器の動作のうちの、一つ若しくはそれ以上のものを含むユーザインプットを含む  
請求項 1 乃至 6 のうちのいずれかに記載のユーザインタフェースシステム。

## 【請求項 8】

操作リクエストが、ユーザにより実施される回転動作である  
請求項 7 に記載のユーザインタフェースシステム。

## 【請求項 9】

上記反対側の各々はデジタルオブジェクトの表示をディスプレイできる  
請求項 1 乃至 8 のうちのいずれかに記載のユーザインタフェースシステム。

## 【請求項 10】

デジタルオブジェクトがオーディオファイルである  
請求項 9 に記載のユーザインタフェースシステム。

## 【請求項 11】

上記表示がデジタルオブジェクトと関連するアイコンである  
請求項 10 に記載のユーザインタフェースシステム。

## 【請求項 12】

デジタル画像を操作するための方法であって、  
ディスプレイ上に、前側と少なくとも一つの反対側を有する、少なくとも一つのデジタル画像をディスプレイするステップ、  
上記ディスプレイ上に上記前側をディスプレイするステップ、  
上記前側と、上記反対側の各々に、少なくとも一つの検出部分を配置するステップ、  
上記検出部分を用いてユーザにより操作リクエストを検出するステップ、及び、  
リクエストが検出されると、ディスプレイされている側が少なくとも部分的に回転して視界から消え少なくとも一つの反対側が少なくとも部分的にディスプレイされるルーティンを開始するステップ  
を含み、  
回転の開始方向はリクエストに依拠して決定される、  
方法。

## 【請求項 13】

上記検出部分は、上記デジタル画像に関して、上記前側と上記反対側の各々のうちの一つの規定された部分である  
請求項 12 に記載の方法。

## 【請求項 14】

操作リクエストがルーティンの実行の間にユーザに維持される場合のみ、完全な回転が維持される  
請求項 13 に記載の方法。

## 【請求項 15】

ルーティンによりルーティンの実行の間にユーザにフィードバックが与えられる  
請求項 12 又は 14 に記載の方法。

10

20

30

40

50

## 【請求項 16】

フィードバックが画像の回転のアニメーションである  
請求項 15 に記載の方法。

## 【請求項 17】

アニメーションが、画像の表面に直交する軸に関して、画像の回転を描く  
請求項 16 に記載の方法。

## 【請求項 18】

操作リクエストが、一つ若しくはそれ以上の手のジェスチャを含むユーザのインプット、及び、ユーザインタフェースシステムに対するスタイラス、ペン、若しくは電子周辺インプットの動作を含む

10

請求項 12 乃至 17 のうちのいずれかーに記載の方法。

## 【請求項 19】

操作リクエストが、ユーザインプットにより実施される回転動作により開始される  
請求項 18 に記載の方法。

## 【請求項 20】

上記反対側の各々はデジタルオブジェクトの表示をディスプレイできる  
請求項 12 乃至 19 のうちのいずれかーに記載の方法。

## 【請求項 21】

デジタルオブジェクトがオーディオファイルである  
請求項 20 に記載の方法。

20

## 【請求項 22】

デジタルオブジェクトが画像ファイルである  
請求項 21 に記載の方法。

## 【請求項 23】

上記表示がデジタルオブジェクトと関連するアイコンである  
請求項 21 又は 22 に記載の方法。

## 【請求項 24】

請求項 1 乃至 11 のうちのいずれかーに記載のユーザインタフェースシステムを実装するコンピュータユニットを含む  
デジタル画像を操作するためのシステム。

30

## 【請求項 25】

請求項 1 乃至 11 のうちのいずれかーに係るユーザインタフェースシステムを制御するための命令を含む  
コンピュータプログラム。

## 【請求項 26】

請求項 12 乃至 23 のうちのいずれかーに記載の方法を実装するコンピュータシステムを制御するための命令を含む  
コンピュータプログラム。

## 【発明の詳細な説明】

## 【技術分野】

40

## 【0001】

本発明は、コンピュータデバイスのディスプレイ上でデジタル画像を操作するためのシステム及び方法に関する。

## 【背景技術】

## 【0002】

コンピュータユーザにとって、コンピュータデータをインプットし操作するプロセスは退屈で困難であることが多い。この理由の一つは、コンピュータインタフェースが非常に人為的であり、所望の結果を達成するのに複雑な操作を要求するからである。

## 【発明の概要】

## 【発明が解決しようとする課題】

50

## 【 0 0 0 3 】

多くのコンピュータユーザは、時間をかけて、インタフェースに精通する。しかしながら、インタフェースに関してどれだけ多くの相互作用を有するのに関わり無く、困難を経験するユーザが依然存在する。例えば、コンピュータに関して殆どトレーニングを経ていないコンピュータユーザ、若しくは、まれにしかコンピュータにアクセスしないユーザには、コンピュータに所望のタスクを実行させるのに何のインプットが要求されるのか思い出すことは、困難である。コンピュータスクリーン上にディスプレイされるデジタル画像などのオブジェクトを操作する場合には、特にそうである。デジタル画像は、多くの抽象化層を有することが多く、精密且つ正確に画像を操作するために、ユーザが多くの層に取り組み更にコンピュータインタフェースに関して適合し緻密であることを要求する。

10

## 【 課題を解決するための手段 】

## 【 0 0 0 4 】

第 1 の形態では、本発明は、  
デジタル画像を操作するためのユーザインタフェースシステムであって、  
少なくとも一つのデジタル画像をディスプレイするように構成されたディスプレイを含み、

上記少なくとも一つのデジタル画像は、前側と、ユーザにディスプレイ可能な少なくとも一つの反対側を有し、前側と、反対側の各々は、少なくとも一つの検出部分を有し、

上記検出部分は、ユーザにより操作リクエストを検出し、該操作リクエストが検出されると、ディスプレイされている側が少なくとも部分的に回転して視界から消え少なくとも一つの反対側が少なくとも部分的にディスプレイされるルーティンを開始するように構成され、

20

回転の開始方向はリクエストに依拠して決定される、  
ユーザインタフェースシステム。

## 【 0 0 0 5 】

ユーザインタフェースシステムの形態では、上記検出部分は、上記前側と上記反対側の各々のうちの一つの規定された部分であってもよい。

## 【 0 0 0 6 】

ユーザインタフェースシステムの形態では、操作リクエストがルーティンの実行の間にユーザに維持される場合のみ、完全な回転が維持されてもよい。

30

## 【 0 0 0 7 】

一つの形態では、ルーティンがルーティンの実行の間にユーザにフィードバックを与えてもよい。

## 【 0 0 0 8 】

一つの形態では、フィードバックが画像の回転のアニメーションであってもよい。

## 【 0 0 0 9 】

一つの形態では、アニメーションが、画像の表面に直交する軸に関して、画像の回転を描いてもよい。

## 【 0 0 1 0 】

一つの形態では、操作リクエストが、手のジェスチャ、及び、スタイラス、ペン、若しくは電子周辺機器の動作のうちの一つであってもよい。

40

## 【 0 0 1 1 】

一つの形態では、操作リクエストが、ユーザインタフェースシステムにインプットされる、手のジェスチャ、スタイラスの動作、ペン、及び電子周辺機器のうちの、一つのものを含むユーザインプットを含んでもよい。

## 【 0 0 1 2 】

一つの形態では、操作リクエストが、ユーザインプットにより実施される回転動作であってもよい。

## 【 0 0 1 3 】

一つの形態では、上記反対側の各々はデジタルオブジェクトの表示をディスプレイでき

50

る。

【 0 0 1 4 】

一つの形態では、デジタルオブジェクトがオーディオファイルであってもよい。

【 0 0 1 5 】

一つの形態では、上記表示がデジタルオブジェクトと関連するアイコンであってもよい。

【 0 0 1 6 】

第2の形態によると、本発明は、

デジタル画像を操作するための方法であって、

ディスプレイ上に、前側と少なくとも一つの反対側を有する、少なくとも一つのデジタル画像をディスプレイするステップ、

上記ディスプレイ上に上記前側をディスプレイするステップ、

上記前側と、上記反対側の各々に、少なくとも一つの検出部分を配置するステップ、

上記検出部分を用いてユーザにより操作リクエストを検出するステップ、及び、

リクエストが検出されると、ディスプレイされている側が少なくとも部分的に回転して視界から消え、少なくとも一つの反対側が少なくとも部分的にディスプレイされるルーティンを開始するステップ

を含み、

回転の開始方向はリクエストに依拠して決定される、

方法である。

【 0 0 1 7 】

第3の形態によると、本発明は、

第1の形態に係るインタフェースを有するコンピュータユニットを含むデジタル画像を操作するためのシステムである。

【 0 0 1 8 】

第4の形態によると、本発明は、

第1の形態に係るユーザインタフェースシステムを制御するための命令を含むコンピュータプログラムである。

【 0 0 1 9 】

第5の形態によると、本発明は、

第2の形態に係る方法を実装するコンピュータシステムを制御するための命令を含むコンピュータプログラムである。

【 0 0 2 0 】

第6の形態によると、本発明は、

第4の形態若しくは第5の形態のうちの一つに係るコンピュータプログラムを格納するコンピュータ読み取り可能媒体である。

【図面の簡単な説明】

【 0 0 2 1 】

本発明の範囲内に含まれ得る他の形態にもかかわらず、添付の図面を参照して、例示としてのみ好適な実施形態を記載する。

【図1】本発明の実施形態を実装するのに利用されるコンピュータシステムの概略図である。

【図2 a】本発明の実施形態に係る、オブジェクトをデジタル画像フリップして添付する方法のステップを示すフロー図である。

【図2 b】本発明の実施形態に係る、オブジェクトをデジタル画像フリップして添付する方法のステップを示すフロー図である。

【図3 A】本発明の実施形態に係る、フリッピングアニメーションの一連のスクリーンショットである。

【図3 B】本発明の実施形態に係る、フリッピングアニメーションの一連のスクリーンシ

10

20

30

40

50

ョットである。

【図３Ｃ】本発明の実施形態に係る、フリッピングアニメーションの一連のスクリーンショットである。

【図３Ｄ】本発明の実施形態に係る、フリッピングアニメーションの一連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】

【００２２】

図１を参照すると、本発明の実施形態に係る方法を実装するように構成されたコンピュータシステム１００が示される。該システム１００は、表面実装ディスプレイ１０２を含む、パーソナルコンピュータ（以下、“テーブルトップコンピュータ”と称する。）の形態である。ディスプレイ１０２は、ユーザインタフェース１０４を表示し、更に、そこに表示されるデジタル画像及び他の情報を操作するためのユーザインプットを受けるように構成されている。デジタル画像は、前側と、テーブルトップコンピュータ１００のユーザにディスプレイ可能な少なくとも一つの反対側を、有する。前側及び反対側は、ユーザにより操作リクエストを検出するように構成された少なくとも一つの検出部分を有する。リクエストを検出すると、テーブルトップコンピュータ１００は、画像の現在ディスプレイされている側が視界から回転して少なくとも一つの反対側がディスプレイされるルーティンを開始する。

【００２３】

上述の機能を達成するために、テーブルトップコンピュータ１００は、マザーボード１１０、中央処理ユニット（ＣＰＵ）１０３、読み取り専用メモリ（ＲＯＭ）１０５、ランダムアクセスメモリ（ＲＡＭ）１０７、及び、ディスクドライブ１０９やイーサネット（登録商標）ポート１１１などのインプット／アウトプットデバイスを、含む。サーバ、パーソナルコンピュータ、端末、無線若しくは携帯式コンピュータデバイスなどの、一つ若しくはそれ以上の他のコンピュータデバイスに接続する複数の通信リンクが設けられてもよい。複数の通信リンクのうちの少なくとも一つが、電話線、若しくは他のタイプの通信リンクを介する外部コンピュータネットワークと接続されてもよい。

【００２４】

テーブルトップコンピュータ１００は、テーブルトップコンピュータ１００に接続するプロジェクタ１２０を含むディスプレイ１０２も含む。プロジェクタ１１７はユーザインタフェース１０４を表面１１９に投影する。ディスプレイ１０２は、ユーザインタフェース１０４上にディスプレイされたオブジェクトを操作するための、ユーザインプットを検出できるセンサ１１８も含む。一つの実施形態では、ユーザインプットは、手動（即ちジェスチャ）及び／又はスタイラス若しくはペンからのインプットを含み得る。ハードウェアに加えて、テーブルトップコンピュータ１００は、ハードディスク上に存在する（マイクロソフトコーポレーションにより供給される、マイクロソフトウインドウ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み、該オペレーティングシステムは、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得る環境を与える。この点において、テーブルトップコンピュータ１００には、ユーザによりインプットされる操作リクエストを検出する検出モジュールが、ロードされる。

【００２５】

本明細書に記載される実施形態では、デジタル画像は、物理的写真と同様に、二つのディスプレイ可能な側を有するデジタル写真の形態である。操作リクエストは、とりわけ、ユーザインタフェースの一つの部分から別の部分へ画像を移動する若しくはドラッグするリクエスト、更には、画像の別の側がユーザに示されるようにデジタル画像を“フリップ”するリクエストを、含む。フリッピング動作を実行するため、ユーザインタフェース１０４及びセンサ１１８と協働するフリッピングモジュールが設けられる。フリッピングリクエストにตอบสนองして、フリッピングモジュールはアニメーションプロセスを実行する。その際、ディスプレイされるビューが隠されデジタル画像の反対側がディスプレイされるようにデジタル画像が一つのビューから回転するかのよう、デジタル画像がディスプレイ

上で動画となる。この回転は、デジタル画像の直線のエッジに直交する軸上のものであってよく、又は、デジタル画像の直線のエッジに直交する軸及び並行の軸の組み合わせ上で回転するものであってもよい。この回転には、物理的なカード、写真、若しくはオブジェクトが手作業で返されるのと同様な、視覚的効果がある。オーディオ若しくはテキストファイル、又は他のデジタルデータなどのファイルを含む（がこれに限定されない）データオブジェクトが、デジタル画像のフリップされる側と関連付けられる、アタッチメントモジュールがテーブルトップコンピュータ 100 に設けられてもよい。この実施形態では、ユーザは、例えば、デジタル画像に、該画像がフリップされると実行され得るメディアファイルをアタッチできる。この機能により、ユーザは、物理的なオブジェクトと相互作用するのに要求されるのと同じ動作及びスキルを用いて、画像を仮想的にフリップできる親しみやすい経験をできる。

10

#### 【0026】

図 2 a を参照すると、本発明の実施形態に係る方法を概説するフローチャートが示される。ステップ 200 では、本発明の実施形態に係るソフトウェアアプリケーションは、フリップリクエストの存在を（周期的に）検出する。フリップリクエストが検出されると、ステップ 202 が呼び出され、フリッピング動作が開始される。ステップ 202 のフリッピング動作が開始されると、フリッピング動作が維持されるかどうかに関して判断がなされる（ステップ 204）。フリッピング動作が維持されるのならば、ステップ 206 が呼び出され、フリッピング動作が完了する。フリッピング動作が完了した後、該方法は、次のファイルリクエストの待機（ステップ 200）に戻る。

20

#### 【0027】

逆に、フリッピング動作が維持されないのならば、該方法はステップ 208 に進み、部分的フリッピング動作が開始する。部分的フリッピング動作が完了すると、該方法は、ステップ 200、即ち、ファイルリクエストの待機に戻る。

#### 【0028】

より詳細には、検出モジュールは、デジタル画像をフリップするユーザインプットリクエストを検出する。フリップリクエストは、手動、及び/又は、スタイラス、ペン、若しくは他の電子周辺デバイスの動作の形態で、ユーザにより与えられる。フリップリクエストを検出するために、検出モジュールは、センサ 118 でユーザインプットタッチポイントを求めてモニタすることにより、ユーザインタフェース 104 上で受けられたユーザインプットタッチポイントの位置を判定する。タッチポイントは、タッチポイントの画像座標の成分の絶対値を判定することにより、計算される。一つ若しくはそれ以上のタッチポイントが、規定された検出部（“ホットスポット”とも言う。）内に位置すると、検出モジュールが判定すると、判定モジュールは、フリッピングモジュールにフリッピング動作を開始するように命令を出す。

30

#### 【0029】

ホットスポットは、ユーザの振る舞い若しくはハードウェア構成に基づいて、構成され得る。一つの実施形態では、ホットスポットはデジタル画像の上左コーナと下右コーナの両方を含む。フリッピング動作を命令するために、ユーザは、親指と中指を用いてホットスポットに同時にタッチし、親指と中指とを寄せるようなつまみ動作を実行する。このように、ユーザのインプットは、ユーザが物理的な写真のひっくり返しをする動作と、実際上同じ動作である。同じ動きはスタイラスやペンを用いても実行され得る。その場合、ユーザは先ずスタイラスを用いてホットスポットをタッチし、スタイラスを持ち上げないで別のホットスポットに向かって動かし、これにより、ひっくり返しのテーブルトップのジェスチャをシミュレートする。この実施形態の実装を示す方法として、限定するものではないが、実装の一つの方法のためのコードのリストが、表 1（付属 1、Annex ture 1）に示される。

40

#### 【0030】

図 2 a を更に参照すると、フリッピング動作 202 のために利用されるコントロールポイントに対応して、フリッピングモジュールが、ユーザインタフェース上でピクセル座標（x

50

、 $y$  ) の引数を受け、該座標は、仮想環境でフリップされるオブジェクトを配置するのに利用されるキャッシュされた逆変換マトリクスを用いて、変換される。 $x$  座標は、フリップの開始方向を判定するのに等しい重み付けを与えるように、 $y$  座標と比較すべく、法線化される。この実施形態の実装を示す方法として、限定するものではないが、実装の一つの方法のためのコードのリストが、表 2 ( 付属 2、Annex ture 2 ) に示される。

#### 【 0 0 3 1 】

ひっくり返しの方向は、デジタル画像の初期制御ポイントの位置により決定される。前に説明したように、各々の制御ポイントはピクセル座標 ( $x$ 、 $y$ ) を有する。ひっくり返しの方向を決定するにあたり、ユーザが一つの制御ポイントのみタッチすれば、個々の  $x$  及び  $y$  成分の絶対値が考慮され、最も大きい成分 ( 即ち、画像の中心から最も大きい距離を伴う成分 ) がひっくり返しの方向を決定する。ユーザがデジタル画像で複数のポイントをタッチする場合には、制御ポイントの全ての座標の平均が先ず計算され、その結果の座標が、最も大きい成分を決定しよってひっくり返しの方向を決定するのに、用いられる。ひっくり返しの距離は ( ポイントが当初 - 0 . 5 から 0 . 5 まで延在する ) 位置成分の 2 倍であり、距離は [ - 1 . 0、1 . 0 ] の範囲となる。ひっくり返しの角度は、距離のアーコサインである。モジュールは、一旦開始されると同じ方向でフリップを継続して、モジュールが逆側にひっくり返しているのか “ 前 ” 側に戻そうとしているのか判定する、という状態を維持する。画像のひっくり返し自体は、引き手順で発生し、このことはユーザインタフェースと連絡してひっくり返し手順をアニメ化する。フィードバックが欠如し、遅延し、非現実的であり若しくは存在しない、従来のコンピュータユーザインタフェースとは異なり、この機能はインタフェースを利用することでユーザの経験を拡張するフィードバックをユーザに与えることを可能にする。言い換えれば、ひっくり返しの方向若しくはフリップの取り消しを検出するというフィードバックを設けることにより、現実の生活環境におけるユーザの物理的な相互作用と将に同様に、ユーザに対してリアルタイムのフィードバックをもたらすことによるインタフェースの利用をユーザに示す、助けとなる。この実施形態の実装を示す方法として、実装の一つの方法のためのコードのリストが、表 3 ( 付属 3、Annex ture 3 ) に示される。

#### 【 0 0 3 2 】

次に図 2 b を参照すると、ひっくり返された画像の “ 背面 ” ( 即ち、他の側 ) にオブジェクトが添付され得る方法が記載されている。ステップ 2 1 0 では、ユーザがアタッチメントリクエストを有するか否かに関して判断される。アタッチメントリクエストが開始されると、システムは進み、画像がひっくり返されたかどうか判断する ( ステップ 2 1 2 ) 。画像がひっくり返されれば、オブジェクトが添付される ( ステップ 2 1 4 ) 。その後、該方法は、アタッチメントリクエストが開始されたかどうかに関する周期的判断に戻る。

#### 【 0 0 3 3 】

逆に、画像がひっくり返されていなければ、オブジェクトに対するアタッチメントをなすことができず、該方法はステップ 2 1 0 に、即ち、アタッチメントリクエストがなされたかどうかに関する周期的判断に戻る。

#### 【 0 0 3 4 】

より詳細にいうと、デジタルオブジェクトがユーザインタフェース 1 0 4 上に移されると、チェックリンク機能が、移動されたオブジェクト上の、タッチポイントの直ぐ下のオブジェクトを選択する。デジタル画像が存在し該画像がフリップモジュールによりひっくり返されたのであれば、デジタルデータと画像との間でアタッチメントが発生する。デジタルオブジェクトがアタッチされると、又は、アタッチメントから解放されると、オブジェクトは、レイアウトアルゴリズムを受ける。2 つの例示のレイアウトアルゴリズムは、 “ Flow Layout ” 及び “ Space - Filling Thumbnails ” レイアウトであるが、同じ “ タイプ ” のオブジェクトを積み重ねてスタックする “ Stacked Layout ” を含むどのレイアウトが用いられてもよい。この実施形態により、ユーザは、メッセージ、情報、クリップ、既述若しくは他のアイテムを画像にアクテ



ィブ化して加えられるので、ユーザインタフェース 104 に関して更にユーザの経験を拡張するものである。アタッチメントは容易にアクセス可能であり、取外し可能であり、コンピュータの技術の無い人にとっても操作可能である。本実施形態の提示のグラフィックディスプレイ及び構造により、オブジェクトの容易なファイル管理がもたらされるからである。アタッチメントの追加の際、若しくは取り外しの際、ユーザは、反復して画像を単に“フリップ”すればよく、これにより容易に格納即ちファイルすることができる。従来のコンピュータインタフェースでは、画像フリッピング、オブジェクトアタッチメント、及びファイリングの全体の処理は、困難で煩わしいタスクである。しかしながら、この記載中の実施形態では、全体の処理は、通常のユーザ（、特に、コンピュータに詳しくないユーザ）が殆ど訓練することなく完了できる通常の直感的な作業で達成し得る。この実施形態の実装を示す方法として、実装の一つの方法のためのコードのリストが、表 4（付属 4、Annex ture 4）に示される。

10

#### 【0035】

##### 例

図 3 A - 図 3 D を参照すると、ユーザが操作したい画像 300 を選択する。ユーザは、あたかもユーザが現実のオブジェクトを操作しているかのように、テーブル上に投影されたデジタル画像 300 にタッチできる。ユーザは、デジタル画像をより近づけてもよく更により離してもよく、ユーザインタフェース 104 うえの新しいどの位置に画像を移動させてもよい。

#### 【0036】

20

デジタル画像 300 をフリップして、ユーザは、規定のエリア 302 を用いてデジタル画像 300 を返してフリッピングジェスチャをシミュレートする。実施形態では、ユーザが画像上に 2 本の指を配置すること、つまり各々の指を近接の写真角に配置することにより、このことは達成される。ユーザは、画像を横切って 2 本の指を引き画像をひっくり返すのであるが、一つの実施形態では、“コーナ”トライアングルはデジタル画像の縁にレンダーされ、“コーナ”トライアングル内に配置されたユーザの指、又はスタイラス若しくはペンは、フリップアクションを開始し、画像を超えて引かれると、該画像は、近接の 2 つの縁を結合する直線に直交する水平軸上で回転する様が見られる。ユーザからのフリップ動作が、画像のひっくり返しの最中に、半分までで取りやめられると、フリッピング動作がキャンセルされたということで、画像はもとの状態に戻る。フリッピング作業中、画像がテーブル上に配置された物理的な写真であるかのように、シミュレートされたターン動作のアニメーションとして、ディスプレイされ得る（図 3 B 及び図 3 C 参照）。

30

#### 【0037】

図 3 D では、ひっくり返された画像が、半透明の白色 304 でレンダリングされており、一方、別の実施形態では、ひっくり返された画像を描くべく異なるグラフ詳細でレンダリングが完了され得る。ディスプレイ上の他のアイテムが、ひっくり返された画像を覆うように引っ張ってこられると、画像は異なる振る舞いをする。画像がひっくり返されると、ユーザは種々のデジタルデータを画像に添付できる。そのようなデータには、オーディオファイル、テキスト、マルチメディアファイル若しくは他のデジタルデータが含まれるが、これらに限定されるものではない。ユーザは、デジタルデータオブジェクトを選択してこのオブジェクトをひっくり返された画像を覆って配置することにより、デジタルデータの添付を開始できる。メニューを利用することにより、若しくは、オブジェクトがひっくり返された画像を覆う際の静止動作により、このことが為され得る。アタッチメントが為されると、数字を伴う円が、ひっくり返された画像の背後に現れる。

40

#### 【0038】

ユーザは、オーディオオブジェクトの一つを記録して、画像にこれらのオブジェクトを添付できるが、これらオブジェクトは画像上に一つの円としてディスプレイされる。画像から該円を引きずり出すことにより、オーディオリンクは消去でき、これによりこのリンクは中断される。

#### 【0039】

50

上述のディスプレイは、投影された画像をディスプレイする表面の形態であったが、当然ながらディスプレイはどんな形態でもよい。例えば、ディスプレイはタッチスクリーンディスプレイの形態でもよいし、単に、キーボードやマウス等の従来のインプットによりディスプレイへのインプットが受けられる従来のディスプレイであってもよい。

【0040】

要求されていないが、図面を参照して記載される実施形態は、開発者の利用のために、アプリケーションプログラミングインタフェース（API）を経由して、若しくは、一連のライブラリとして、実装されてもよく、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、携帯式コンピュータデバイスのオペレーティングシステムなどの、別のソフトウエアアプリケーション内に含まれてもよい。一般に、プログラムモジュールは、特定機能を実施する若しくは特定機能の実施をアシストする、ルーティン、プログラム、オブジェクト、コンポーネント、及び、データファイルを含むのであり、当然ながら、ソフトウエアアプリケーションの機能は、複数のルーティン、オブジェクト及びコンポーネントに分散されて、既述の実施形態及び本明細書に記載の広範な発明と同じ機能が達成され得る。それらの変更及び修正は、当業者の視野の範囲内である。

【0041】

【表1】

## ANNEXURE 1

Procedure for deciding whether to initiate a flip

```
1 bool Image::flipSelected() {  
2     GLfloat x = fabsf(currMPos.x), y = fabsf(currMPos.y);  
3     return !inBlackhole &&  
4         (y - x > RConfig::CORNER_PROP ||  
5          x - y > 0.5*aspect - 0.5 + RConfig::CORNER_PROP);  
6 }
```

【0042】

10

20

## 【表 2】

## ANNEXURE 2

Procedures for flipping an object, given the current contact position on the screen

```

1  bool Image::flipto(int screen_x, int screen_y) {
2      P3<GLfloat> avgPos;
3      //determine in texture coords where the control point is
4      setObject(avgPos, screen_x, screen_y, true);
5      avgPos.x /= aspect; //normalise between 0 and 1
6       //(we always flip along the bigger direction)
7
8      float cosval, dist;
9      if (!flipping) { //not already flipping
10         xflip = fabs(avgPos.y) > fabs(avgPos.x);
11         dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
12         backflip = dist < 0;
13         wasflipped = flipped;
14         flipping = true;
15     } else {
16         dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
17     }
18
19     //now the flipping state toggles if the distance has changed in sign
20     flipped = wasflipped != (backflip != dist < 0);
21
22     //determine the angle to rotate the image
23     cosval = fabs(dist);
24     if (cosval < 1.0) {
25         flipangle = acosf(cosval); //between 0 and 90
26
27         //now, if we started on the "wrong" side, we rotate the other way
28         //we also do a magical flip at the centre if flipped
29         if (xflip == backflip)
30             flipangle *= -1.0;
31
32         //now we are here, whenever we have a change from
33         // the start, we want to subtract from 180 degrees
34         if (flipped != wasflipped)
35             flipangle = -flipangle;
36
37     } else {
38         //dist is > 1.0, so touch point moved away — don't flip
39         flipangle = 0.0;
40     }
41
42     pc.makeDirty();
43     return true;
44 }

```

【 0 0 4 3 】

【表 3】

## ANNEXURE 3

```

45 void Image::drawPartialFlip() {
46     glPushMatrix();
47     glRotatef(flipangle*RAD2DEG,
48         xflip ? 1.0 : 0.0,
49         xflip ? 0.0 : 1.0,
50         0.0);
51     if (flipped) {
52         drawReverse();
53         //also draw the back if we are flipping
54         if (flipping) {
55             glRotatef(180,
56                 xflip ? 1.0 : 0.0,
57                 xflip ? 0.0 : 1.0,
58                 0.0);
59             glCallList(*model);
60         }
61     } else {
62         drawTexture();
63         if (flipping) {
64             glRotatef(180,
65                 xflip ? 1.0 : 0.0,
66                 xflip ? 0.0 : 1.0,
67                 0.0);
68             drawReverse(); /* if rendering ?? */
69         }
70     }
71 }
72 glPopMatrix();
73 }

```

10

20

【 0 0 4 4 】

## 【表 4】

## ANNEXURE 4

Procedure, called each time an object is moved, to check whether it should be attached to another object

10

```

1 void Resource::checkLink() {
2     // get the topmost object in the environment at the 'touch' location
3     P3<int> scr(pc.getScreen() + clickPos);
4     PResource top = env->selectTop(scr.x, scr.y, this, true);
5
6     PResource cur = 0; // current parent, if any
7     if (link)
8         cur = link->getParent();
9
10    if (cur != top && canDetach()) { //if changed
11        // remove old dependencies, if any
12        removeDependenciesOnParent();
13
14        if (top && top->isFlipped()) {
15            if (canLink(top) && top->canLink(this)) {
16                Layout *lay = top->getChildLayout();
17                /* if new parent has a layout manager, add to that */
18                if (lay) {
19                    lay->add(this);
20                    lay->layout();
21                    link = lay;
22                } else { /* else use 'basic' linking method */
23                    link = PDAdapter::link(this, top);
24                }
25                pushFlasher(top); // give feedback
26            }
27        }
28    }
29 }
```

20

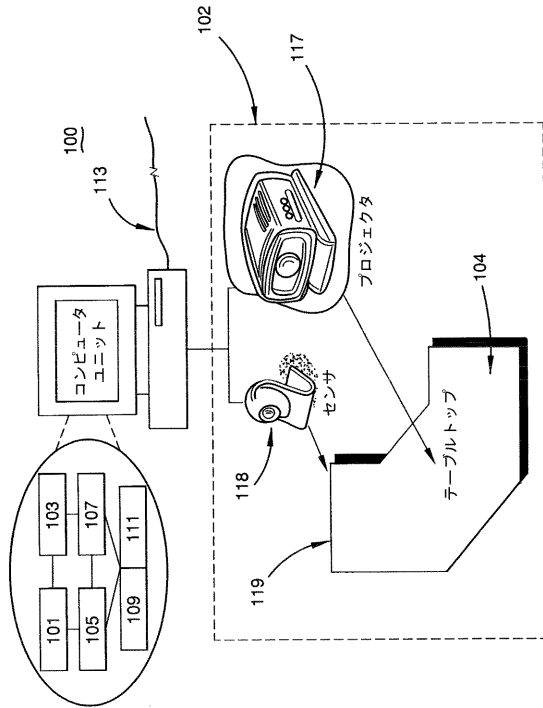
30

## 【符号の説明】

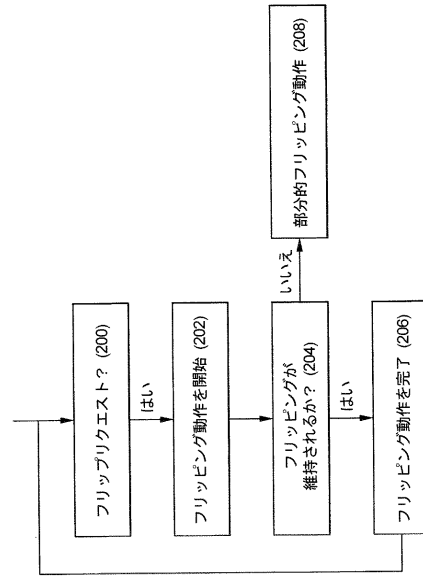
## 【 0 0 4 5 】

1 0 0・・・コンピュータシステム、1 0 2・・・ディスプレイ、1 0 4・・・ユーザインタフェース（テーブルトップ）、1 1 7・・・プロジェクタ、1 1 8・・・センサ。

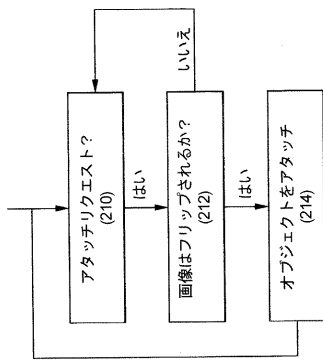
【図 1】



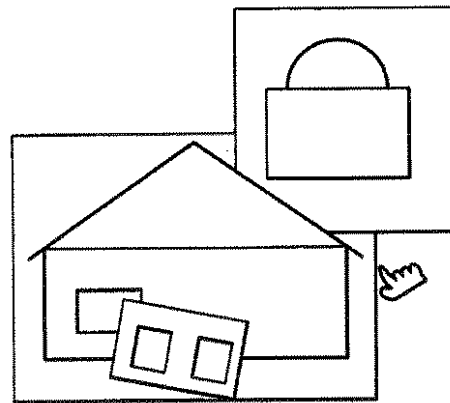
【図 2 a】



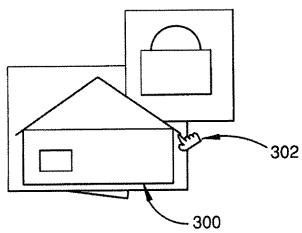
【図 2 b】



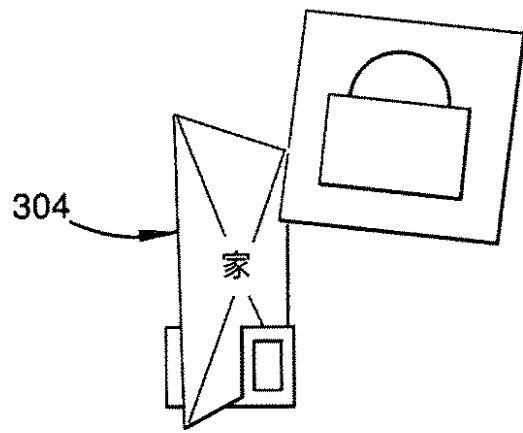
【図 3 B】



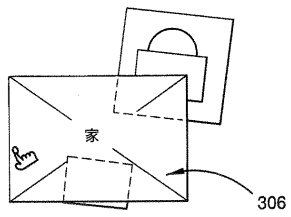
【図 3 A】



【図 3 C】



【図 3 D】



---

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 高瀬 勤

(56)参考文献 特開 2 0 0 4 - 0 2 1 5 2 2 ( J P , A )

特開 2 0 0 2 - 2 8 8 6 9 0 ( J P , A )

国際公開第 2 0 0 7 / 0 9 1 2 9 8 ( W O , A 1 )

特開 2 0 0 5 - 0 6 3 3 2 8 ( J P , A )

特開 2 0 0 4 - 2 5 9 0 6 5 ( J P , A )

特開 2 0 0 7 - 1 5 6 4 0 8 ( J P , A )

(58)調査した分野(Int.Cl. , DB名)

G 0 6 F 3 / 0 4 8 1

J S T P l u s ( J D r e a m I I I )