

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6619798号
(P6619798)

(45) 発行日 令和1年12月11日(2019.12.11)

(24) 登録日 令和1年11月22日(2019.11.22)

(51) Int.Cl.

F 1

G06F 3/0488 (2013.01)
G06F 3/0484 (2013.01)G06F 3/0488 130
G06F 3/0484 150

請求項の数 12 (全 18 頁)

(21) 出願番号 特願2017-508035 (P2017-508035)
 (86) (22) 出願日 平成27年9月29日 (2015.9.29)
 (65) 公表番号 特表2017-532634 (P2017-532634A)
 (43) 公表日 平成29年11月2日 (2017.11.2)
 (86) 國際出願番号 PCT/US2015/052755
 (87) 國際公開番号 WO2016/053916
 (87) 國際公開日 平成28年4月7日 (2016.4.7)
 審査請求日 平成30年8月29日 (2018.8.29)
 (31) 優先権主張番号 14/500,997
 (32) 優先日 平成26年9月29日 (2014.9.29)
 (33) 優先権主張国・地域又は機関
米国(US)

(73) 特許権者 314015767
マイクロソフト テクノロジー ライセンシング、エルエルシー
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ
 (74) 代理人 100079108
弁理士 稲葉 良幸
 (74) 代理人 100109346
弁理士 大貫 敏史
 (74) 代理人 100117189
弁理士 江口 昭彦
 (74) 代理人 100134120
弁理士 内藤 和彦

最終頁に続く

(54) 【発明の名称】低レイテンシのインクレンダリングパイプライン

(57) 【特許請求の範囲】

【請求項 1】

ディスプレイデバイス上にインクコンテンツをレンダリングするための方法であって、
インク入力アクションを受信するためのアクションコンテキストを定義すること、
アプリケーションによって入力アクションを受信することであって、前記入力アクションは入力コンテキストを有すること、

前記入力コンテキストは前記アクションコンテキストによって指定された基準を満たすため、前記入力アクションをインク作成アクションとして検出すること、

前記インク作成アクションをウェットインクレンダリングスレッドに導くことであって、前記ウェットインクレンダリングスレッドは、前記アプリケーション用のアプリケーション処理スレッドとは異なること、

前記ウェットインクレンダリングスレッドによって、前記インク作成アクションコンテキストをレンダリングしてウェットインクコンテンツを生成すること、

ディスプレイに前記ウェットインクコンテンツを出力すること、

インク作成の終わりに対応する追加の入力アクションを検出すること、

前記インク作成の終わりに対応する前記追加の入力アクションを検出した後、前記ウェットインクコンテンツを中間インクコンテンツに変換すること、

前記中間インクコンテンツを前記アプリケーションレンダリングスレッドに転送すること、

前記アプリケーションレンダリングスレッドによる前記中間インクコンテンツの前記レ

10

20

ンダリングと、前記ウェットインクレンダリングスレッドによってレンダリングされた前記ウェットインクコンテンツの前記除去と、を同期化すること、

前記インク作成アクションをアプリケーション処理スレッドに転送すること、

前記アプリケーション処理スレッドによって、前記インク作成アクションをレンダリングして、ドライインクコンテンツを生成すること、及び、

前記ウェットインクレンダリングスレッドによってレンダリングされた前記中間インクコンテンツを除去するとともに、前記ドライインクコンテンツをディスプレイに出力すること、

を含む、方法。

【請求項 2】

10

前記アクションコンテキストは、モードコンテキストを含み、

前記モードコンテキストは、ペン入力、スタイルス入力、タッチ入力、マウス入力、又はそれらの組み合わせの、入力モードを含む、請求項 1 に記載の方法。

【請求項 3】

前記アクションコンテキストは、位置コンテキストを更に含む、請求項 1 に記載の方法。

【請求項 4】

前記位置コンテキストは、前記ディスプレイデバイスの表示域の或る領域を含み、前記領域は前記表示域の全域よりも小さい、請求項 3 に記載の方法。

20

【請求項 5】

前記位置コンテキストは、前記ディスプレイデバイスの表示域の複数の領域を含む、請求項 3 に記載の方法。

【請求項 6】

前記位置コンテキストは、複数の表示域からの複数の領域を含む、請求項 3 に記載の方法。

【請求項 7】

前記入力コンテキスト及び前記アクションコンテキストのうちの少なくとも 1 つは、前記ウェットインクレンダリングスレッドによってレンダリングされる前記ウェットインクコンテンツに関する 1 つ以上の表示プロパティを更に含む、請求項 1 に記載の方法。

30

【請求項 8】

前記ウェットインクレンダリングスレッドによる前記インク作成アクションの前記レンダリングの間、前記ウェットインクレンダリングスレッドによってレンダリングされる前記ウェットインクコンテンツに関する変更された 1 つ以上の表示プロパティを形成するために、前記 1 つ以上の表示プロパティが更新されたことを検出すること、及び、

前記変更された 1 つ以上の表示プロパティを使用して、前記インク作成アクションの少なくとも一部をレンダリングすること、

を更に含む、請求項 7 に記載の方法。

【請求項 9】

前記 1 つ以上の表示プロパティが更新されたことの検出に先立ってレンダリングされた前記ウェットインクコンテンツの少なくとも一部は、前記インク作成の終わりに対応する前記追加の入力アクションの検出に先立って前記変更された 1 つ以上の表示プロパティを用いてレンダリングされる、請求項 8 に記載の方法。

40

【請求項 10】

前記アクションコンテキストは、インク作成コンテキストを更に含む、請求項 1 に記載の方法。

【請求項 11】

プロセッサと、コンピュータ記憶メモリと、を備えるコンピューティングシステムであつて、

前記コンピュータ記憶メモリは、前記プロセッサによって実行された時に、ディスプレ

50

イデバイス上でインクコンテンツをレンダリングするための方法を実装する、コンピュータ実行可能命令をその上に記憶しており、

前記方法は、

インク入力アクションを受信するためのアクションコンテキストを定義すること、

アプリケーションによって入力アクションを受信することであって、前記入力アクションは入力コンテキストを有すること、

前記入力コンテキストは前記アクションコンテキストによって指定された基準を満たすため、前記入力アクションをインク作成アクションとして検出すること、

前記インク作成アクションをウェットインクレンダリングスレッドに導くことであって、前記ウェットインクレンダリングスレッドは前記アプリケーション用のアプリケーション処理スレッドとは異なること、

前記ウェットインクレンダリングスレッドによって、前記インク作成アクションをレンダリングしてウェットインクコンテンツを生成すること、

ディスプレイに前記ウェットインクコンテンツを出力すること、

インク作成の終わりに対応する追加の入力アクションを検出すること、

前記インク作成の終わりに対応する前記追加の入力アクションを検出した後、前記ウェットインクコンテンツを中間インクコンテンツに変換すること、

前記中間インクコンテンツを前記アプリケーションレンダリングスレッドに転送すること、

前記アプリケーションレンダリングスレッドによる前記中間インクコンテンツの前記レンダリングと、前記ウェットインクレンダリングスレッドによってレンダリングされた前記ウェットインクコンテンツの前記除去と、を同期化すること、

前記インク作成アクションをアプリケーション処理スレッドに転送すること、

前記アプリケーション処理スレッドによって、前記インク作成アクションをレンダリングして、ドライインクコンテンツを生成すること、及び、

前記ウェットインクレンダリングスレッドによってレンダリングされた前記中間インクコンテンツを除去するとともに、前記ドライインクコンテンツをディスプレイに出力すること、を含む、

コンピューティングシステム。

【請求項 1 2】

前記入力コンテキスト及び前記アクションコンテキストのうちの少なくとも 1 つは、前記ウェットインクレンダリングスレッドによってレンダリングされる前記ウェットインクコンテンツに関する 1 つ以上の表示プロパティを更に含む、請求項 1 1 に記載のコンピューティングシステム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

[0001] タブレットコンピューティングデバイス（又は同様のタイプの組み込み型デバイス）は、しばしば、ユーザがデータをデバイスに入力できるようにするタッチスクリーンを含む。ユーザによるデータ入力は、指、マウス、スタイラス（又はペン）などを使用して実行することができる。ペン（又は指）からデバイスへの入力は「インク」と呼ばれる。

【0 0 0 2】

[0002] インクは、コンピューティングデバイスによって処理及び操作できるようにデジタル化され、表示される。これは、典型的にはデジタル化されたインクをデバイス上のオペレーティングシステムに送信し、プロセッサにインクを処理させた後、処理されたインクをグラフィックスカードに送信することによって実行される。次いでグラフィックスカードは、（ユーザのアクションの効果である）インクをモニタ又はディスプレイ上にレンダリングする。

【0 0 0 3】

10

20

30

40

50

[0003] 現在のインクレンダリング技法の1つの欠陥は、導入されるレイテンシである。インクのレンダリングは遅れる傾向がある一方で、ペン及び指は高速で事実上瞬間的な傾向がある。実際、インクレンダリングプロセス全体で、約70から200ミリ秒或いはそれよりも長い可能性のあるレイテンシを導入する可能性がある。インクレンダリングにおけるこのレイテンシが、低速で厄介なユーザ体験を生み出す可能性がある。

【0004】

[0004] 特にこのレイテンシは、オブジェクトを直接操作する知覚的バリアを破るという点で、望ましくない影響である。もちろんこれらは、コロケート体験が存在する状況ではほとんどが知覚される。言い換えれば、入力及び出力が互いにコロケートしている場合(タブレットコンピューティングデバイス又はタッチスクリーン上のスタイルスによる書き込みなど)、レイテンシの影響は最大である。

10

【0005】

[0005] ユーザがスタイルス、マウス、指(又は他のタッチ入力)、或いは、表示域内にインクコンテンツにとって望ましい位置をトレースするための他の描画デバイスを用いてインクコンテンツを作成する間、インクの表示に関するレイテンシを向上させるためのシステム及び方法が提供される。ユーザがスタイルス/マウス/タッチ入力/他のデバイスを使用することによって作成されるインクコンテンツの表示のための時間を削減又は最小化するために、オペレーティングシステム内で他のアプリケーションスレッドと並行して動作する、個別のインクレンダリングプロセススレッドを使用することができる。アプリケーション内でインクコンテンツを作成することが望ましい場合、インクコンテンツの作成に対応するユーザ対話は、個別のインクレンダリングプロセススレッドによって扱うことができる。これにより、プロセスフロー内で他のイベントを扱うアプリケーションに起因する、インクコンテンツを表示する際の潜在的な遅延を回避することができる。個別のインクレンダリングプロセススレッドがアプリケーションと通信するため、個別のインクレンダリングプロセスによってレンダリングされるインクコンテンツは、依然としてアプリケーションのコンテキストを使用してレンダリングすることができる。これにより、インクコンテンツを対応するアプリケーションに転送した後に、インクコンテンツが有し得る外観とほぼ同様の外観で、インクコンテンツをユーザに表示することが可能となる。

20

【発明の概要】

【0006】

30

[0006] 本概要は、以下の発明を実施するための形態でより詳細に説明する概念の選択を、簡略化された形で紹介するために提供される。本概要は、請求される課題の主要な特徴又は不可欠な特徴を識別することは意図されておらず、請求される課題の範囲を決定する際に、単独で、補助として使用することも意図されていない。

【0007】

[0007] 本発明は、添付の図面を参照しながら以下で詳細に説明する。

【図面の簡単な説明】

【0008】

【図1】[0008]ユーザによって生成されるウェットインクコンテンツをレンダリングするための、コンポーネント間の関係についてのフロー図を概略的に示す図である。

40

【図2】[0009]様々なタイプのインクコンテンツをレンダリングするための、コンポーネント間の関係についてのフロー図を概略的に示す図である。

【図3】[0010]インクコンテンツをレンダリングするための、コンポーネント間の関係についてのフロー図を概略的に示す図である。

【図4】[0011]様々なタイプのインクコンテンツのレンダリングに関するプロセスフローを示す図である。

【図5】[0011]様々なタイプのインクコンテンツのレンダリングに関するプロセスフローを示す図である。

【図6】[0011]様々なタイプのインクコンテンツのレンダリングに関するプロセスフローを示す図である。

50

【図7】[0012]プロセススレッド間でのインクコンテンツの転送に関するプロセス通信フローを示す図である。

【発明を実施するための形態】

【0009】

[0013] 様々な態様において、ユーザがスタイルス、マウス、指（又は他のタッチ入力）、或いは、表示域内でインクコンテンツにとって望ましい位置をトレースするための他の描画デバイスを用いてインクコンテンツを作成する間、インクの表示に関するレイテンシを向上させるためのシステム及び方法が提供される。ユーザがスタイルス／マウス／タッチ入力／他のデバイスを使用することによって作成されるインクコンテンツの表示のための時間を削減又は最小化するために、オペレーティングシステム内で他のアプリケーションスレッドと並行して動作する、個別のインクレンダリングプロセススレッドを使用することができる。アプリケーション内でインクコンテンツを作成することが望ましい場合、インクコンテンツの作成に対応するユーザ対話は、個別のインクレンダリングプロセススレッドによって扱うことができる。これにより、プロセスフロー内で他のイベントを扱うアプリケーションに起因する、インクコンテンツを表示する際の潜在的な遅延を回避することができる。個別のインクレンダリングプロセススレッドがアプリケーションと通信するため、個別のインクレンダリングプロセスによってレンダリングされるインクコンテンツは、依然としてアプリケーションのコンテキストを使用してレンダリングすることができる。これにより、インクコンテンツを対応するアプリケーションに転送した後に、インクコンテンツが有し得る外観とほぼ同様の外観で、インクコンテンツをユーザに表示することが可能となる。10

【0010】

[0014] 様々な態様において、本明細書で説明するシステム及び方法は、プロセススレッド間でのインクコンテンツの取り扱いの遷移中、ユーザによって作成される新しいインクコンテンツを連続的に表示することができる。新しいインクコンテンツの所与の部分の作成が完了したものと見なされる場合、完了したコンテンツのレンダリングを、対応するアプリケーションについてのプロセススレッドに転送することができる。インクコンテンツの特定の部分の作成が完了しているため、新しいインクコンテンツに関するレイテンシの難点はもはや問題ではない。結果として、完了したコンテンツの表示におけるレイテンシについての問題が低減又は最小化された対応するアプリケーションのプロセススレッドによって、更なるレンダリングを実行することができる。ユーザ体験を更に向上させるために、インクレンダリングスレッドから対応するアプリケーションプロセススレッドへの、完了したコンテンツの表示の転送を、同期化することができる。これにより、両方のプロセススレッドがインクコンテンツの表示を試行する期間をなくす一方で、コンテンツをほぼ連続して表示できるようになる。本発明の態様は、いずれのプロセスもインクコンテンツの表示を試行していない状況を防止することもできる。言い換えれば、本発明の態様において、一方又は他方のプロセスはインクコンテンツを表示することができるが、両方が同時に表示することはない。20

【0011】

[0015] ユーザによって作成されたインクコンテンツをレンダリングする際の難点の1つは、ユーザのアクションが結果としてインクコンテンツを表示させているように見えるように、インクコンテンツを十分迅速に表示することである。ユーザは、約50ミリ秒又はそれ以下のように短いインクコンテンツの作成のためのユーザアクションに応答して、遅延を知覚できることが予め決定されている。ユーザによって作成されるインクコンテンツの表示における遅延が目に見える場合、コンテンツ作成はもはや、書き込み面上で物理的なペンを使用してインクを作成する感覚を有さない可能性があるため、遅延はユーザ体験を混乱させる可能性がある。残念なことに、インクコンテンツが作成されるアプリケーションについて、プロセススレッドを使用してインクコンテンツの表示を試行することは、望ましくないレイテンシ期間につながる可能性がある。これは、典型的なアプリケーションが、アプリケーションによって実行される他のアクションに関するアプリケーションブ30

ロセススレッド内で、インクコンテンツに関するアクションを優先順位付けする機能を有さないという事実に起因する。

【0012】

[0016] 代替として、インクコンテンツのレンダリングは、オペレーティングシステムから完全に分離させることができる。これにより、オペレーティングシステムをバイパスし、代わりに新しいインクを表示域内に直接表示する、ハードウェア及び／又はソフトウェアの機能を使用して、ユーザによって作成される新しいインクコンテンツを表示することが可能になる。新しいインクコンテンツを作成した後、新しいインクコンテンツはオペレーティングシステムに渡され、ここでは、対応するアプリケーションによって提供されるコンテキスト内にインクコンテンツがレンダリングされるように、表示機能に対して任意の修正を適用することができる。このタイプの方法は、新しいインクコンテンツの表示におけるレイテンシを低減させることができるが、新しいインクコンテンツの初期の表示は、コンテンツの最終レンダリングと異なる場合がある。10

【0013】

[0017] いくつかの態様において、本明細書で説明するシステム及び方法の利点は、インクコンテンツの表示に関するレイテンシを低減させるために、コンピュータの性能を向上させることである。こうした向上は、インクコンテンツ作成中のユーザ体験を向上させることにもつながり得る。

【0014】

[0018] 様々な態様において、本明細書で説明するシステム及び方法は、スタイルス、タッチ入力を受信するためのタッチインターフェース、マウス、又は、表示域内のカーソルの動きを制御可能な別のデバイスなどの、入力デバイスを使用して、ユーザがインク入力を作成できるようにすることが可能である。インクコンテンツを作成したいというユーザの望みは、ユーザのアクションに関するコンテキストに基づいて決定又は検出可能である。インクコンテンツの作成のためのアクションに対応するアクションコンテキストが定義可能である。次いで、ユーザアクションに関する入力コンテキストを検出し、入力アクションがインク作成アクションであるかどうかを判別するために、アクションコンテキストと比較することが可能である。20

【0015】

[0019] 1つのタイプのコンテキストは、入力デバイスの性質自体とすることができます。これはモードコンテキストと呼ぶことが可能であり、スタイルスと表面との対話、指又は別の形状とタッチ入力のための表面との対話、マウスを使用したカーソルアイコン（又は他のカーソル識別子）の移動、及び／又は、任意の他の便利なデバイスに基づいて、検出可能である。30

【0016】

[0020] 第2のタイプのコンテキストは、位置コンテキストとすることができます。任意選択で、ディスプレイの表示域全体、タッチ表面、入力パッド、又は別のタイプのユーザインターフェースは、いつでもインクコンテンツを受信できる準備が整っている。このタイプのオプションでは、ユーザのアクションは、モードコンテキストのみに基づいて、及び／又は、位置コンテキストとは異なる他のタイプのコンテキストに基づいて、インク作成アクションとして検出可能である。代替として、ユーザインターフェース内（ユーザインターフェースの表示域内など）の1つ以上の領域を、インクコンテンツが受信可能な領域として識別可能である。このタイプのオプションでは、ユーザアクションは、モードコンテキストと位置コンテキストとの組み合わせに基づいて、インク作成アクションとして検出される。潜在的にインクコンテンツを受信するための領域は、デバイス用のオペレーティングシステムによって事前定義すること、デバイス上で実行中のアプリケーションによって事前定義すること、又は、ユーザにより1つ以上の領域に関する定義に基づいて作成することが可能である。40

【0017】

[0021] 第3のタイプのコンテキストは、インク作成コンテキストとすることができます。

50

1つのオプションは、選択されたモードコンテキスト及び／又は位置コンテキストを有する任意のユーザアクションを、インク作成アクションとして解釈することであり得る。代替として、ユーザアクションがインク作成アクションである旨を決定することは、インクコンテンツの作成を開始するための入力デバイスを用いるシングルタップ／クリック又はダブルタップ／クリックなどの、特定タイプのユーザアクションを検出することに依存可能である。こうした指定されたアクションは、インク作成コンテキストに対応可能である。インク作成コンテキストは、ユーザが事前にインク作成を開始したかどうかに応じて変化し得ることに留意されたい。例えば、ユーザによる入力デバイス又はカーソルの移動などの追加のアクションは、インク作成が開始された後、及び、デバイスを入力表面から離して持ち上げるなどのインク作成を終わらせるアクションが検出されるまで、インク作成アクションと解釈することができる。10

【0018】

[0022] 本発明の態様は、前述の3つのコンテキストに限定されない。他のコンテキストが可能である。

【0019】

[0023] モードコンテキスト、位置コンテキスト、アクションコンテキスト、及び／又はユーザアクションがインク作成アクションであるかどうかの判別に使用される任意の他のタイプのコンテキストに基づいて、ユーザ入力アクションがインク作成アクションであるかどうかを判別するために、ユーザ入力アクションをコンテキストに照らしてテストすることができる。入力アクションがインク作成アクションである場合、ユーザによって作成されているインクコンテンツをレンダリングするために、入力アクションを、インクレンダリングスレッドなどの個別のプロセススレッドにルーティングすることができる。コンテキスト条件を満たしていないアクションは、ユーザ入力アクションを扱うための従来のプロセススレッドでの処理のために、アプリケーション及び／又はオペレーティングシステムに渡すことができる。入力アクションのテストは、入力アクションの性質を決定するために短い遅延を発生させる可能性があるが、インクコンテンツがユーザによって作成された場合にこれをレンダリングするために個別のプロセススレッドを使用することに基づいて、レイテンシの大幅な（数十ミリ秒などの）低減を達成することができる。このタイプのテストは、「ヒットテスト」と呼ばれることがある。本発明の一態様において、ヒットテストは発生しない。代わりに、接触に基づいてインクが生成され、ヒットテストなしで（及び結果として生じる遅延なしで）表示される。入力がインクコンテキストを満たさないように変化した場合、インクプロセスは取り消され、表示されたウェットインクは除去される。20

【0020】

[0024] 基礎を成すアプリケーションとは別のインクレンダリングプロセススレッドにおいてインクコンテンツを扱うことに伴う潜在的な問題の1つは、新しいインクコンテンツが最終的に基礎を成すアプリケーションに転送（又は、基礎を成すアプリケーションと同期）されなければならないことである。様々な態様において、表示されるインクコンテンツにおける変化（表示の一時的喪失を含む）を低減又は最小化しながら、及び／又は、両方のプロセススレッドによるコンテンツの2重表示を低減又は最小化しながら、この転送を実行させることができ可能なシステム及び方法が提供される。これにより、デバイスの性能及び結果として生じるユーザ体験の両方において、利益をもたらすことができる。30

【0021】

定義

[0025] 以下の考察において、本発明に従ったシステム及び方法の例は、Microsoft（商標）の動作環境に基づく動作環境を用いて例示することができる。加えて、ある用語は、様々な態様を説明する際に言及し易いように用いられる。以下の明示的定義は、様々な態様の理解を助けるために与えられる。

【0022】

[0026] ウェットインク：インクストロークが進行中の間に、すなわち、接触（例えば、40

50

ペン)が表面からまだ持ち上げられていない間にレンダリングされるインク。様々な態様において、ウェットインクはセミドライインク又はドライインクに変換されるか、或いは置き換えられることが可能である。変換プロセスは、ペンが表面から持ち上げられた後、しばらく続行可能である。レンダリングされたインクは、セミドライインク又はドライインクに置き換えられるまでウェットインクのままである。様々な態様において、ウェットインクは個別のインクレンダリングスレッド上にレンダリングされる。

【0023】

[0027] ドライインク：インクストロークが完了した後にレンダリンク又は再レンダリングされるインク、或いは、入力以外のソース(残留インクコンテンツ)からロードされたインクデータからレンダリングされるインク。

10

【0024】

[0028] セミドライインク：ドライインク層への同期を保留している中間層内のインク。様々な態様において、セミドライインクは、個別のインクレンダリングプロセススレッド上にレンダリング可能である。

【0025】

[0029] ウェット／ドライ同期化：ドライインクをレンダリングし、ドライインクスレッド上で可視化することが可能であり、並びに、a) 対応するドライインクが可視化された時に、ウェットインクが依然として可視化されないこと、b) 対応するドライインクが可視化される前に、ウェットインクが消失しないこと、及びc) インクスレッドが、入力の処理及び更なるウェットインクのレンダリングから決してブロックされないことを、保証できるように、ウェットインクをインクスレッド上でクリアすることが可能な、メカニズム。

20

【0026】

[0030] ホスト：内部で新しいインクコンテンツの一部がユーザによって作成される、基礎を成すアプリケーション又はフレームワーク。

【0027】

[0031] ホストスレッド：基礎を成すアプリケーション又はユーザインターフェースのためのプロセススレッド。

【0028】

[0032] インクレンダリングスレッド：選択されたタイプの入力が受信され、ウェットインクをレンダリングするために処理される、背景スレッド(好ましくはオペレーティングシステム内に作成される)。このスレッドは、プロセス内でウェットインクをレンダリングするためのすべてのインスタンスによって共有される。

30

【0029】

[0033] ドライインクスレッド：その上でインクデータがレンダリングコンポーネント又はスレッドに送達される、スレッド。一態様において、ホストスレッドはレンダリングスレッドであるが、専用インクレンダリングスレッドを含む他のオプションが可能である。

【0030】

[0034] 入力リダイレクション：デフォルトではUIスレッドの代わりに指定されたスレッドに自動的にルーティングされる入力合致指定特徴(ウェットインクの作成のための入力など)を有するように、入力スタックに登録するためのメカニズム。様々な態様において、指定特徴は、スタイルス、指(又は他のタッチ入力)、又はウェットインクの作成に好適な他の入力デバイスの使用などの、入力のモダリティに対応可能である。様々な態様において、指定特徴は、ウェットインク入力を受信するために画定されたユーザインターフェースの一部分におけるこうした入力デバイスの使用を、更に指定することができる。

40

【0031】

[0035] InkManager: Windowsランタイムクラス。1つ以上のインクオブジェクトの入力、操作、及び処理(手書き文字認識を含む)を管理するためのプロパティ及び方法を提供する。

【0032】

50

[0036] D 2 D : Direct 2 D。2次元ジオメトリ、ビットマップ、及びテキストに対して高性能及び高品質のレンダリングを提供する、ハードウェア加速、即時モードの2次元グラフィクスAPI。Direct 3 Dなどの他のレンダリング技術との相互運用をサポートするように設計される。

【0033】

[0037] D 3 D : Direct 3 D。レンダリングパイプラインを用いてプリミティブを描画するため、又は、コンピュートシェーダを用いて並列操作を実行するための、低水準API。異なるGPU実装をコヒーレント抽象化の背後に隠し、グラフィクス特有のプロセッサを駆動するように設計される。

【0034】

[0038] D Comp : Direct Composition。変形、効果、及びアニメーションを用いて高性能ビットマップコンポジションを実行可能にする、Windowsコンポーネント。様々なソースからのビットマップを組み合わせ、スクリーンにレンダリングするためにデスクトップウインドウマネージャ(DWM)を活用することができる。

【0035】

[0039] DXGI : Microsoft DirectXグラフィクスインフラストラクチャ。Direct 3 Dグラフィクスランタイムから独立可能な低水準タスクを管理する。いくつかのバージョンのDirect 3 Dに対する共通フレームワークも提供する。

【0036】

インクコンテンツを扱うためのプロセスフロー

10

[0040] 図1は、個別のインクレンダリングプロセススレッドを使用してウェットインクコンテンツをレンダリングするためのプロセスフローの例を示す。図1に示されるプロセスフローにおいて、ユーザは、マウス、スタイルス、ペン、又はタッチインターフェース(タッチ入力を受信するため)などのハードウェアコンポーネントを使用して、ウェットインクコンテンツを作成するための入力アクションを提供することができる。入力ハードウェア102はこの入力をハードウェアドライバ104に渡し、最終的に入力テスタ110に渡すことができる。入力テスタは、入力アクションがウェットインクコンテンツを作成するためのアクションに対応するかどうかを判別するために、ハードウェアの性質及び/又は入力アクションに関連付けられたディスプレイ内の位置を含む、入力アクションについてのコンテキストを評価することができる。

20

【0037】

[0041] ユーザアクションがウェットインクコンテンツの作成に対応する場合、入力アクションはウェットインクプロセススレッド120へと迂回される。図1に示されるように、ウェットインクプロセススレッド120は限定数のアクションのみを扱い、これによって、ウェットインクレンダリングアクションの取り扱いにおける遅延の潜在性を低減又は最小化することができる。これは、汎用ユーザインターフェース及び/又はアプリケーションプロセススレッド130によって扱われる複数タイプのアクションとは対照的である。例えば、インクコンテンツの取り扱いのみに関して、ユーザインターフェース及び/又はアプリケーションプロセススレッド130は、インクコンテンツの一般的な表示のためのアプリケーションプログラムインターフェース132を提供すること、手書き文字認識などのインクコンテンツの追加処理のためのアプリケーションプログラムインターフェース134を提供すること、結果として生じるインクコンテンツを表示できるように、インクストロークが追加された場合に関する入力136を受信すること、及び、シリアル化されたインクコンテンツなどの事前に作成されたドライインクコンテンツ138の表示を扱うことが、可能である。

30

【0038】

[0042] ウェットインク入力を受信した後、ウェットインクプロセススレッド120はウェットインクコンテンツをレンダリング124し、レンダリングされたコンテンツを、コンポジタ150などのオペレーティングシステムのグラフィクス処理要素に戻すことができる。最終的に、レンダリングされたウェットインクは、ユーザに表示するためにハード

40

50

ウェアグラフィクスディスプレイ要素 160 に渡される。

【0039】

[0043] 図 1 に示されるようなプロセスフローを使用するシステムは、様々な利点を提供することができる。例えば、このタイプのプロセスフローは、アプリケーション / UI スレッドから独立している、すなわち、入力が高優先度の背景スレッド（図 1 に示されるウェットインクスレッド）に直接流れ、UI / app スレッドアクティビティによって決してブロックされない、低レイテンシのウェットインクレンダリングを提供することができる。任意選択であるが好ましいことに、システムは、デフォルトのペンタイプで良好に働くウェットベジェアルゴリズム、並びに、従来システムに対して 15 ms 又はそれ以上レイテンシを低減させるための予測を、使用することができる。こうしたシステムは、インク特徴を変更するための圧力モデリングにも適合可能である。加えて、オペレーティングシステム内にあるウェットインクプロセススレッドに部分的に基づき、ウェットインクプロセススレッドは、インク描画属性（色、形状、サイズ、タイプなど）に合致するペンカーソルをレンダリングすることができる。10

【0040】

[0044] 図 2 は、ウェットインク（すなわち、ユーザによる作成プロセス中のインク）及びドライインク（すなわち、ユーザによって予め入力された、及び / 又はデータソースから取得された、インクコンテンツ）の両方を表示するためのプロセススレッドを含む、別のプロセスフローを示す。図 1 のプロセスフローにおいて、ユーザによって作成されたインクコンテンツは、インクコンテンツの作成が完了した後、ウェットインクプロセススレッドによってレンダリング可能である。図 1 におけるドライインクコンテンツのレンダリングは、ソースファイルから取り出されるコンテンツなどの、予め定義されたドライインクコンテンツのレンダリングに対応する。図 2 において、ユーザによって作成されたインクコンテンツを、ウェットインクプロセススレッドから、ドライインクコンテンツのレンダリングのためにユーザインターフェース / アプリケーション / 他のスレッドへ転送できるようにするための、追加のフローが示される。20

【0041】

[0045] 図 2 において、ウェットインクスレッド 120 は、両方のスレッドと通信するコンポーネントを介して、ユーザインターフェーススレッド 130（及び、オプションの個別ドライインクレンダリングスレッド 233）と通信することができる。これらのコンポーネントは、ウェットインクスレッド 120 からユーザインターフェーススレッド 130 へのウェットインクの転送を同期化する、Ink Sync コンポーネント 242 を含むことができる。コンポーネントは、転送が完了したコンポーネントに対して検証を与えるコミットマネージャ 246 も含むことができる。例えばコミットマネージャ 246 は、ウェットインクスレッド 120 によるウェットインクから中間インク（又は、セミウェットインク）への変換が完了した時点を、ユーザインターフェーススレッド 130 に通知することができる。同様に、コミットマネージャ 246 は、ユーザインターフェーススレッド 130 が転送されたドライインクコンテンツのレンダリングを開始した時点を、ウェットインクスレッド 120 に通知することができるため、ウェットインクスレッド 120 は対応する中間インクコンテンツのレンダリングを停止することができる。30

【0042】

コンポーネント関係

[0046] 図 3 は、ユーザ作成のインクコンテンツ（ウェットインク）のレンダリング、並びに、個別のインクレンダリングスレッドからアプリケーションに関連付けられたプロセススレッドへのウェットインクの転送に使用可能な、様々なコンポーネント間の関係を示す。図 3 において、ホストアプリケーション 310 又は他のフレームワーク（潜在的にオペレーティングシステムを含む）は、1 つ以上のアプリケーションプログラミングインターフェース 320 を介して、「Direct Ink Core」モジュール 330 全体と通信することができる。図 3 に示されるアプリケーションプログラミングインターフェース 320 は、入力インターフェース及びインク取り扱いインターフェースを含む。ウェッ4050

トイント入力（又は、ウェットインクプロセススレッドによって扱われることになる別のタイプの入力）として検出された入力がホスト 310 によって受信された場合、入力はモジュール 330 に渡される。

【0043】

[0047] DirectInk Core 実装オブジェクト 332 は、モジュール 330 内に示される他のオブジェクトのほとんどを作成及び調整する、コンテナオブジェクトである。マーシャラ 342 は、インクスレッドを所有し、スレッド間での通知のためのメカニズムを実装し、通知メカニズムを他のコンポーネントから隔離する。入力コンテキストオブジェクト 336 は、入力リダイレクションを登録し、インクスレッド上の入力を受信する、入力シンクオブジェクトである。InkSync オブジェクト 346 は、ウェット／ドライインク同期化のためのメカニズムを実装する、同期化エンジンである。コミットマネージャ 352 は、DComp コミットを実行し、他のコンポーネントを DComp コミット完了通知の詳細から隔離する、共有オブジェクトである。レンダリングマネージャ 370 は、ウェット、セミドライ及びドライレイヤに使用されるレンダラインスタンスを追跡し、レンダリング要求を適切なレンダラインスタンスにルーティングする。レンダマネージャ 370 は、必要に応じてデフォルトのレンダラインスタンスを作成することも可能であり、必要に応じてカスタムレンダラを管理する。レンダラ 372 は、高水準レンダラオブジェクトである。レンダラ実装 374 は、表面／コンテキスト、インクデータ、及び描画属性が与えられた場合、実際のレンダリングを実行する低水準レンダラオブジェクトである。インクモデル 376 は、任意選択で予測を用いて、入力から平滑なベジェセグメントを作成する。10

【0044】

[0048] インクツリーデータオブジェクト 340 は、ウェット、セミドライ、及びドライの層について関連付けられたインクデータと共に、DirectInk Core インスタンスについてのインクサブツリーを表す DComp 視覚サブツリーを作成及び維持する、データオブジェクトである。20

【0045】

[0049] 図 4 は、ウェットインクに対応する入力の到達、及びウェットインク入力のレンダリングを扱うための、データフローを示す。図 4 において、ウェットインクであると識別される入力が、システム入力スタック 410 によって入力コンテキストコンポーネント 420 へと送達される。入力は、任意選択で、任意の入力プラグイン 425 に渡すことができる。入力は InkSync コンポーネント 430 にも渡される。InkSync 430 は、新しいウェットインクコンテンツが作成されている旨の情報を、インクツリーデータコンポーネント 440 に渡すことができる。InkSync は、ウェットインクをレンダリングするよう、レンダマネージャ 450 に命じることもできる。最後に、インクコンテンツの入力が完了した時点で、InkSync 430 はコミットマネージャ 460 に通知することができる。30

【0046】

[0050] 図 5 は、ユーザによるインクコンテンツ部分の入力が完了した後、セミドライ又は中間インクのレンダリングに関するデータフローを示す。図 5 において、データフローは、インクコンテンツ部分の作成が完了したため、インクコンテンツ部分をウェットインク状態から中間インク状態へと移動できる旨を、InkSync コンポーネント 430 がインクツリーデータ 440 に通知した時点で開始される。次いで、レンダマネージャ 450 は、中間又はセミドライのインクレンダラインスタンスを使用して、インクコンテンツの完了部分をレンダリングするための命令を受信可能であり、ウェットインクレンダラインスタンスは、別のインクコンテンツ作成イベントの準備を整えることができる。コミットマネージャ 460 に通知した後、ホスト又は基礎を成すアプリケーションによって管理されるドライインクレンダープロセススレッドに転送するために、インクが使用可能である旨の通知を、マーシャラ 470 に渡すことができる。40

【0047】

50

[0051] 図 6 は、ウェットインクプロセススレッドからドライインクプロセススレッドへのインクの（同期化された）転送に関するプロセスフローを示す。図 6において、Ink Sync 430 は、マーシャラ 470 からインク利用可能信号を受信する。次いで、Ink Sync 430 は、セミドライ状態からドライ状態へインクコンテンツをコピーするよう、インクツリーデータ 440 に通知する。インクをドライインクとしてレンダリングするためのストローク情報は、インクコールバック 672 を使用してインクハンドラ 680 に渡される。インクコンテンツは、ドライインクレンダラインスタンスを用いてレンダリングするために、レンダマネージャ 450 にも渡される。これが完了した後、Ink Sync 430 は、セミドライバージョンのインクコンテンツを非表示にすることができる旨をインクツリーデータ 440 に通知するため、ウェットインクスレッドはそれ以上インクコンテンツを扱わないことになる。10

【0048】

[0052] 図 7 は、ウェットインクレンダリングスレッド（インク）とドライインクレンダリングスレッド（ホスト）についての DComp インスタンス間での通信フローを示す。図 7 における通信フローは、インクコンテンツの表示の不連続性を回避しながら、ドライインクスレッド及びウェットインクスレッドの両方がインクコンテンツの表示を試行する時間を低減又は最小化するために、ウェットインクのドライインクへの転送を如何にして同期化させるかを示す。

【0049】

[0053] 図 7において、インクスレッド 703 は、ウェットからセミウェットへの移動プロセス 710 を開始することによって、呼を始動する。DComp コンポーネント 701 は、コミット呼 712 を用いてインクスレッドに確認する。次いで、インク使用可能呼 714 がホストスレッド 706 に渡される。ホストスレッド 706 は、インクツリー内でデータを移動させるために、セミドライからドライへのコピー呼 716 を始動する。このコピーが完了した後、セミドライデータを非表示にするための呼 718 も実行される。セミドライからドライへの遷移は、コミット呼 722 を用いて DComp ホストスレッド 708 に確認される。次いで、セミドライインクからドライインクへの転送が完了した旨の呼 724 が、インクスレッド 703 に戻される。次いで、確認するためのコミット呼 732 を用いて、ウェットインクスレッドによるセミドライインクのレンダリングがクリアされる 726。次いで、セミドライインクのクリアに関する呼 734 がホストスレッド 706 に渡される。20

【0050】

追加例

[0054] 本発明の様々な実施形態の概要を簡単に説明してきたが、次に、本発明を実行するためには適な例示的動作環境を説明する。本発明の実施形態は、コンピュータ、或いは携帯情報端末又は他のハンドヘルドデバイスなどの他の機械によって実行される、プログラムモジュールなどのコンピュータ実行可能命令を含む、コンピュータコード又は機械使用可能命令の一般的なコンテキストで説明可能である。一般に、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含むプログラムモジュールは、特定のタスクを実行するか又は特定の抽象データ型を実装するコードを指す。本発明は、ハンドヘルドデバイス、家庭用電子機器、汎用コンピュータ、より専門的なコンピューティングデバイスなどを含む、様々なシステム構成内で実施可能である。本発明は、通信ネットワークを介してリンクされたりモート処理デバイスによってタスクが実行される、分散型コンピューティング環境内でも実施可能である。40

【0051】

[0055] 好適なコンピューティングデバイスは、以下のデバイス、すなわちメモリ、1つ以上のプロセッサ、1つ以上の提示コンポーネント、入力／出力（I/O）ポート、I/O コンポーネント、及び電源を、直接的又は間接的に結合するバスを含むことができる。バスは、何が1つ以上のバス（アドレスバス、データバス、又はそれらの組み合わせ）であり得るかを表す。明快にするために様々なコンポーネントが個々のコンポーネントとし50

て記載しているが、実際には、様々なコンポーネントを図で表現することはそれほど明快ではなく、隠喩的には、線はより正確にはグレイカフファジーとなる。例えばディスプレイデバイスなどの提示コンポーネントは、I/Oコンポーネントであると見なすことができる。加えて、多くのプロセッサはメモリを有する。こうしたカテゴリー間で、「ワークステーション」、「サーバ」、「ラップトップ」、「ハンドヘルドデバイス」、「タブレット」などは、すべてが「コンピューティングデバイス」の範囲内で企図されるため、区別されない。

【0052】

[0056] コンピュータ記憶媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、又は他のデータなどの情報を記憶するための任意の方法又は技術で実装される、揮発性及び不揮発性、取り外し可能及び取り外し不可の媒体を含む。コンピュータ記憶媒体は、限定されないが、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)、電気的消去可能プログラマブル読み取り専用メモリ(EEPROM)、フラッシュメモリ又は他のメモリ技術、CD-ROM、デジタル多用途ディスク(DVD)又は他のホログラムメモリ、磁気カセット、磁気テープ、磁気ディスク記憶又は他の磁気記憶デバイス、或いは、所望のデータをエンコードするために使用可能であり、コンピューティングデバイス100によってアクセス可能な、任意の他の媒体を含む。実施形態において、コンピュータ記憶媒体は、フラッシュメモリのような有形のコンピュータ記憶媒体から選択可能である。これらのメモリ技術は、データを瞬間的、一時的、又は永続的に記憶することができる。コンピュータ記憶は通信媒体を含まず、除外する。コンピュータ記憶媒体は、非一時的であり、伝播型データ信号を除外する。10

【0053】

[0057] 他方で、通信媒体は典型的には、コンピュータ可読命令、データ構造、プログラムモジュール、又は他のデータを、搬送波などの変調データ信号又は他の移送メカニズム内で具体化し、任意の情報送達媒体を含む。「変調データ信号」という用語は、その特徴のうちの1つ以上が信号内の情報をエンコードするような方法で設定又は変更された信号を意味する。例を挙げると、限定ではないが、通信媒体は、ワイヤードネットワーク又はダイレクトワイヤード接続などのワイヤード媒体と、音波、RF、赤外線、及び他のワイヤレス媒体などのワイヤレス媒体とを含む。20

【0054】

[0058] メモリは、揮発性及び/又は不揮発性メモリの形のコンピュータ可読媒体を含むことができる。メモリは、取り外し可能、取り外し不可、又はそれらの組み合わせとすることができます。例示的ハードウェアデバイスは、ソリッドステートメモリ、ハードドライブ、光ディスクドライブなどを含む。コンピューティングデバイスは、メモリ又はI/Oコンポーネントなどの様々なエンティティからデータを読み取る、1つ以上のプロセッサを含むことができる。提示コンポーネントは、ユーザ又は他のデバイスに対してデータ指示を提示する。例示的提示コンポーネントは、ディスプレイデバイス、スピーカ、印刷コンポーネント、振動コンポーネント、などを含む。30

【0055】

[0059] I/Oポートは、コンピューティングデバイスを、そのうちのいくつかが組み込み可能なI/Oコンポーネントを含む他のデバイスに論理的に結合できるようにするものである。実例となるコンポーネントは、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信用アンテナ、スキャナ、プリンタ、ワイヤレスデバイスなどを含むことができる。I/Oコンポーネントは、タッチ入力(例えば、キーボード又はマウスとの対話に基づくディスプレイ上の間接的な移動とは対照的に、ディスプレイインターフェースにタッチするか又は接近する)、ジェスチャ入力、触覚入力、音声入力、近接入力、上記で識別された入力デバイスなどの2次入力デバイスとの対話、又は、任意の他の便利なタイプの入力などの、複数の入力タイプから1つ以上の入力タイプを受信する、コンポーネントを含むことができる。40

【0056】

10

20

30

40

50

[0060] コンピューティングデバイスは無線を含むことができる。無線は無線通信を送信及び受信する。コンピューティングデバイスは、様々なワイヤレスネットワークを介して受信される通信及び媒体に適合されたワイヤレス端末とすることができる。コンピューティングデバイス 1100 は、符号分割多元アクセス（「CDMA」）、グローバルシステムフォーモバイル（「GSM」）、又は時分割多元アクセス（「TDMA」）、並びにその他などの、他のデバイスと通信するためのワイヤレスプロトコルを介して、通信可能である。無線通信は、短距離接続、長距離接続、又は、短距離及び長距離の両方のワイヤレス通信接続の組み合わせとすることができます。「短」と「長」タイプの接続に言及する場合、2つのデバイス間の空間的関係を指すことは意味していない。代わりに通常は、異なるカテゴリ又はタイプの接続（すなわち、1次接続及び2次接続）として、短距離及び長距離に言及する。短距離接続は、802.11プロトコルを使用する WLAN 接続などの、ワイヤレス通信ネットワークへのアクセスを提供するデバイス（例えばモバイルホットスポット）への Wi-Fi（登録商標）を含むことができる。別のコンピューティングデバイスへの Bluetooth 接続が、短距離接続の第 2 の例である。長距離接続は、CDMA、GPRS、GSM、TDMA、及び 802.16 プロトコルのうちの 1つ以上を使用する接続を含むことができる。

【0057】

[0061] 本発明の実施形態を、すべての点で制限的ではなく例示的であることが意図される特定の実施形態に関して説明してきた。当業者であれば、本発明がその範囲を逸脱することなく関連する代替の実施形態が明らかとなろう。

【0058】

[0062] 実施形態 1。ディスプレイデバイス上にインクコンテンツをレンダリングするための方法であって、インク入力アクションを受信するためのアクションコンテキストを定義することであって、アクションコンテキストは少なくともモードコンテキストを含む、定義すること、アプリケーションによって入力アクションを受信することであって、入力アクションは入力コンテキストを有する、受信すること、インク作成アクションとして受信される入力アクションを検出することであって、受信される入力アクションは、定義されたアクションコンテキストに対応する入力コンテキストに基づいてインク作成アクションとして検出され、検出されたインク作成アクションはインクコンテンツに対応する、検出すること、検出されたインク作成アクションをウェットインクレンダリングプロセススレッドに導くことであって、ウェットインクレンダリングスレッドはアプリケーション用のアプリケーション処理スレッドとは異なる、導くこと、ウェットインクレンダリングスレッドによって、レイテンシ閾値未満のレイテンシを伴うインクコンテンツをレンダリングすること、及び、インク作成の終わりに対応する入力アクションを検出することを含む、方法。

【0059】

[0063] 実施形態 2。実施形態 1 の方法であって、インクコンテンツを、ウェットインクレンダリングスレッドからアプリケーションレンダリングスレッドへ転送することであって、アプリケーションレンダリングスレッドは、ウェットインクレンダリングスレッドとは異なり、任意選択でアプリケーション処理スレッドとは異なる、転送すること、アプリケーションレンダリングスレッドによって転送されたインクコンテンツをレンダリングすること、及び、ウェットインクレンダリングスレッドによってレンダリングされたインクコンテンツを除去することを、更に含む。

【0060】

[0064] 実施形態 3。実施形態 2 の方法であって、インクコンテンツをウェットインクレンダリングスレッドからアプリケーションレンダリングスレッドへ転送することは、インク作成の終わりに対応する入力アクションを検出した後、インクコンテンツを中間インクコンテンツに変換すること、中間インクコンテンツをアプリケーションレンダリングスレッドに転送すること、及び、アプリケーションレンダリングによって転送された中間インクコンテンツのレンダリングと、ウェットインクレンダリングスレッドによってレンダリ

10

20

30

40

50

ングされたインクコンテンツの除去と、を同期化することを含む。

【0061】

[0065] 実施形態4。上記実施形態のうちのいずれかの方法であって、レイテンシ閾値は、約75ミリ秒又はそれ以下、或いは約60ミリ秒又はそれ以下、或いは約50ミリ秒又はそれ以下である。

【0062】

[0066] 実施形態5。上記実施形態のうちのいずれかの方法であって、モードコンテキストは、ペン入力、スタイラス入力、タッチ入力、マウス入力、又はそれらの組み合わせの、入力モードを含む。

【0063】

[0067] 実施形態6。上記実施形態のうちのいずれかの方法であって、定義されたアクションコンテキストは、位置コンテキストを更に含む。

【0064】

[0068] 実施形態7。実施形態6の方法であって、位置コンテキストはディスプレイデバイスの表示域の或る領域を含み、領域は表示域の全域よりも小さい。

【0065】

[0069] 実施形態8。実施形態6又は7の方法であって、位置コンテキストはディスプレイデバイスの表示域の複数の領域を含むか、又は位置コンテキストは複数の表示域からの複数の領域を含む。

【0066】

[0070] 実施形態9。上記実施形態のうちのいずれかの方法であって、入力コンテキスト及びアクションコンテキストのうちの少なくとも1つは、ウェットインクレンダリングスレッドによってレンダリングされるインクに関する1つ以上の表示プロパティを更に含む。

【0067】

[0071] 実施形態10。請求項9の方法であって、ウェットインクレンダリングスレッドによるインクコンテンツのレンダリングの間、ウェットインクレンダリングスレッドによってレンダリングされるインクに関する1つ以上の表示プロパティにおける変更を検出すること、及び、変更された1つ以上の表示プロパティを使用して、インクコンテンツの少なくとも一部をレンダリングすることを、更に含む。

【0068】

[0072] 実施形態11。実施形態10の方法であって、1つ以上の表示プロパティにおける変更の検出に先立ってレンダリングされたインクコンテンツの少なくとも一部は、インク作成の終わりに対応する入力アクションの検出に先立って変更された1つ以上の表示プロパティを用いてレンダリングされる。

【0069】

[0073] 実施形態12。上記実施形態のうちのいずれかの方法であって、アクションコンテキストはインク作成コンテキストを更に含む。

【0070】

[0074] 上記から、本発明は、明白であり構造に固有である他の利点と共に、上記に記載されたすべてのエンド及びオブジェクトを達成するように、適切に適合されていることがわかるであろう。

【0071】

[0075] ある機能及びそれらの下位組み合わせは実用的であり、他の機能及び下位組み合わせに関係なく採用可能であることを理解されよう。これは特許請求の範囲によって企図され、その範囲内にある。

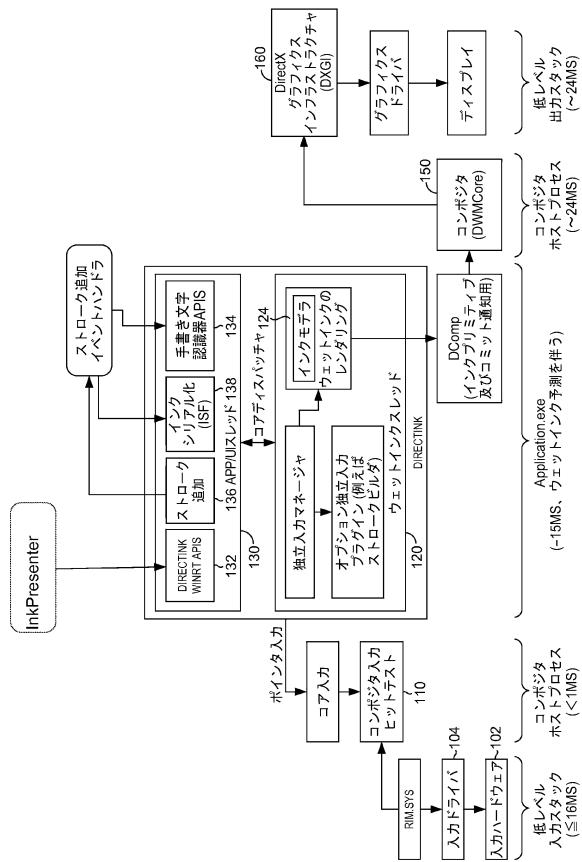
10

20

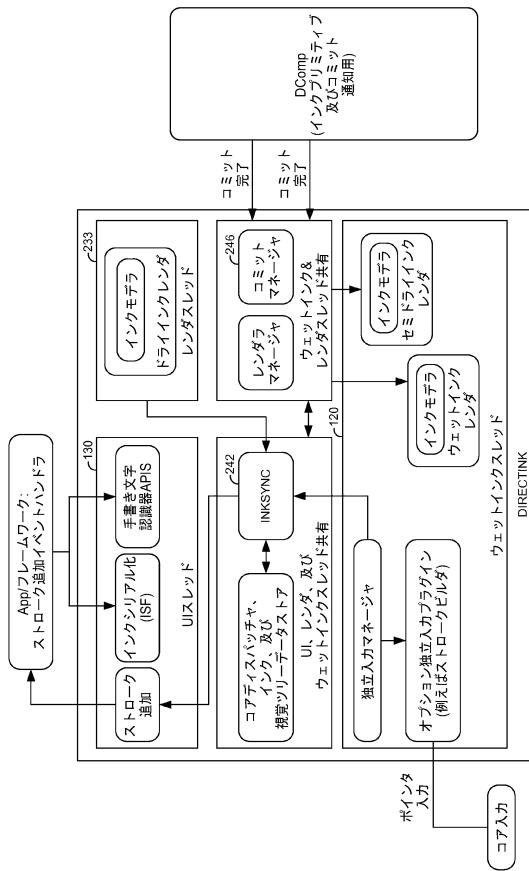
30

40

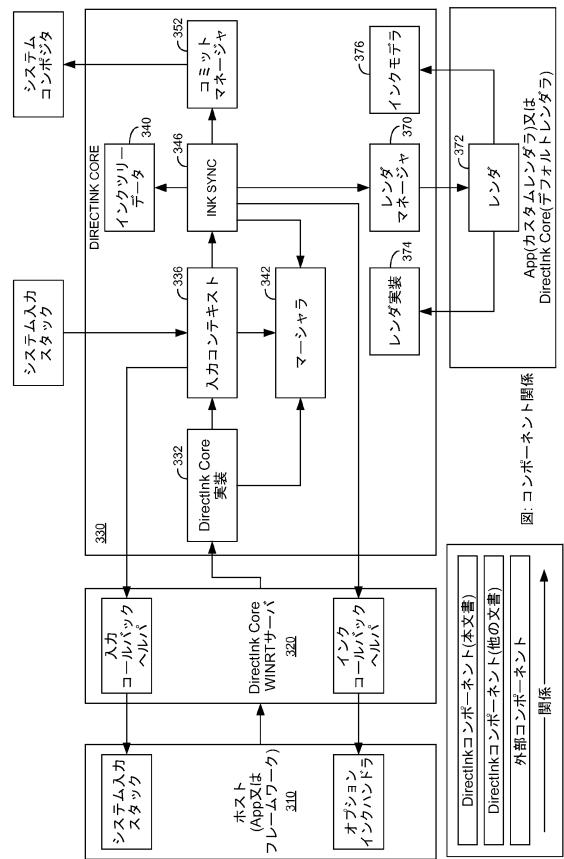
【図1】



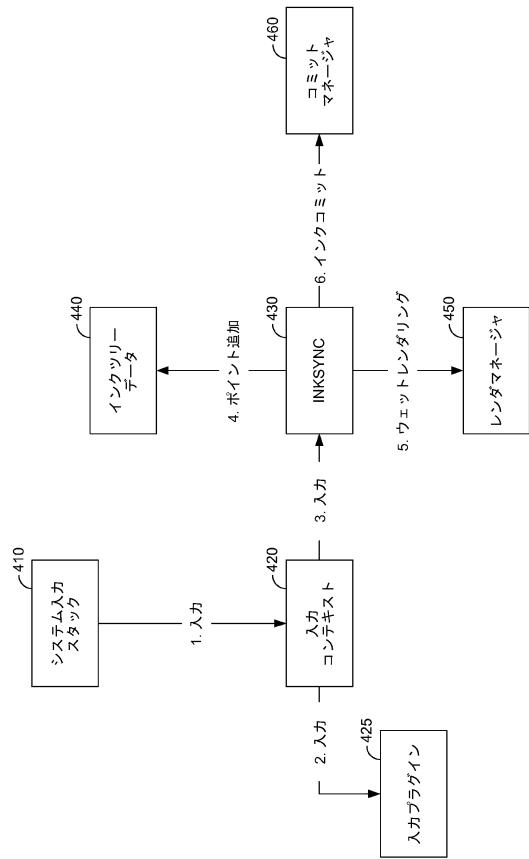
【図2】



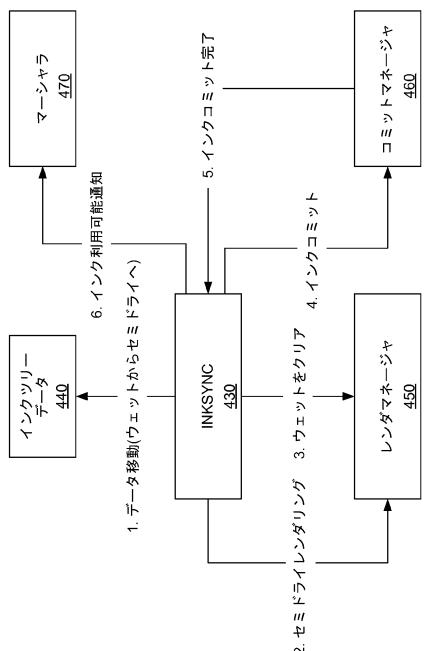
【図3】



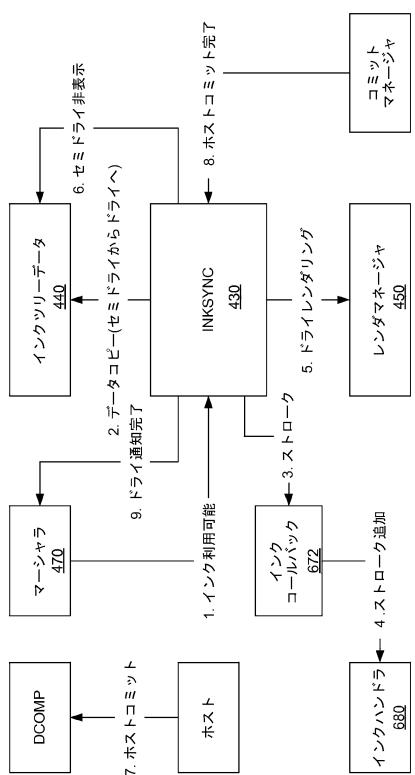
【図4】



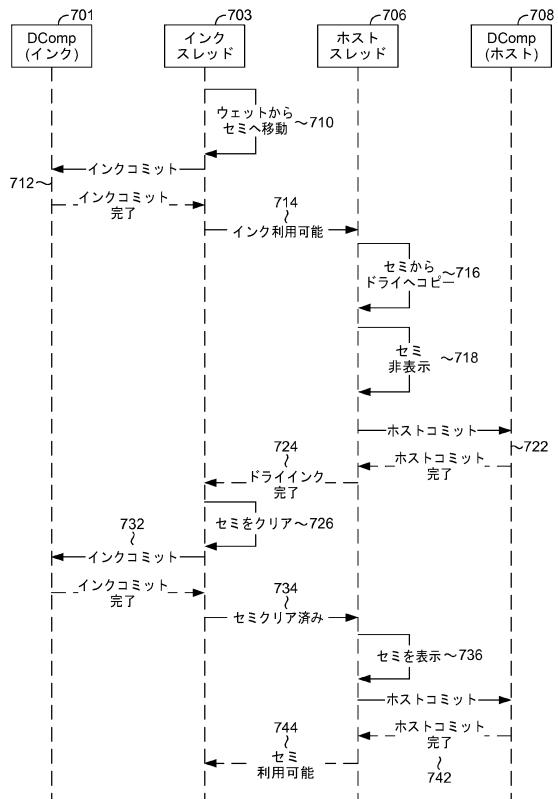
【 四 5 】



【図6】



【図7】



フロントページの続き

(74)代理人 100108213

弁理士 阿部 豊隆

(74)代理人 100140648

弁理士 佐藤 宏樹

(72)発明者 チュ, ジャオ

アメリカ合衆国, ワシントン州 98052-6399, レッドモンド, ワン マイクロソフト
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エーティーティーエヌ
: パテント グループ ドケティング (ビルディング 8/1000)

(72)発明者 メノン, クリシュナン

アメリカ合衆国, ワシントン州 98052-6399, レッドモンド, ワン マイクロソフト
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エーティーティーエヌ
: パテント グループ ドケティング (ビルディング 8/1000)

(72)発明者 シヨン, フェイ

アメリカ合衆国, ワシントン州 98052-6399, レッドモンド, ワン マイクロソフト
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エーティーティーエヌ
: パテント グループ ドケティング (ビルディング 8/1000)

(72)発明者 ホン, ソン

アメリカ合衆国, ワシントン州 98052-6399, レッドモンド, ワン マイクロソフト
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エーティーティーエヌ
: パテント グループ ドケティング (ビルディング 8/1000)

(72)発明者 デューホン, デイビッド ウォーカー

アメリカ合衆国, ワシントン州 98052-6399, レッドモンド, ワン マイクロソフト
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エーティーティーエヌ
: パテント グループ ドケティング (ビルディング 8/1000)

審査官 岩橋 龍太郎

(56)参考文献 米国特許出願公開第2014/0168096(US, A1)

特表2007-509413(JP, A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 3 / 041 - 3 / 0489

G 06 F 3 / 03

G 06 T 1 / 00

G 06 T 11 / 00 - 11 / 40

G 06 T 11 / 60 - 13 / 80

G 06 T 15 / 00 - 17 / 00

G 06 T 17 / 05

G 06 T 17 / 10 - 17 / 30

G 06 T 19 / 00 - 19 / 20