(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5281660号
(P5281660)

(45)発行日　平成25年9月4日(2013.9.4)　(24)登録日　平成25年5月31日(2013.5.31)

(51)Int.Cl.		F I		
G06T 15/80	(2011.01)	G06T	15/80	
G06T 1/20	(2006.01)	G06T	1/20	B
G09G 5/00	(2006.01)	G09G	5/00	555A
G09G 5/36	(2006.01)	G09G	5/36	520Z
		G09G	5/36	530C

請求項の数 17　(全 15 頁)　最終頁に続く

(21)出願番号　特願2010-549691 (P2010-549691)
(86)(22)出願日　平成21年1月30日(2009.1.30)
(65)公表番号　特表2011-517803 (P2011-517803A)
(43)公表日　平成23年6月16日(2011.6.16)
(86)国際出願番号　PCT/US2009/032611
(87)国際公開番号　WO2009/111119
(87)国際公開日　平成21年9月11日(2009.9.11)
審査請求日　平成23年12月21日(2011.12.21)
(31)優先権主張番号　12/041,951
(32)優先日　平成20年3月4日(2008.3.4)
(33)優先権主張国　米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
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
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行

最終頁に続く

(54)【発明の名称】宣言型プレゼンテーション・フレームワークのためのシェーダーベースの拡張

(57)【特許請求の範囲】
【請求項１】
シェーダーの動作を制御するためのシステムであって、
宣言型プレゼンテーション・フレームワークにおいてシェーダーのインスタンス化及び動作を制御する宣言型言語の宣言文が使用されることを可能にする宣言型のプログラミング・モデルを備え、前記シェーダーはグラフィック処理装置上で実行するように動作可能であり、前記宣言型のプログラミング・モデルは、ソフトウェア・アプリケーションにおけるグラフィック効果のために前記グラフィック処理装置を利用するため、前記ソフトウェア・アプリケーションが前記宣言型のプログラミング・モデルを使用することを可能にし、前記宣言型のプログラミング・モデルはブラシが前記シェーダーへの入力として使用されることを可能にするように動作可能であるシステム。
【請求項２】
前記宣言型言語は拡張マークアップ言語（ＸＭＬ）である請求項１に記載のシステム。
【請求項３】
前記宣言型言語は拡張アプリケーションマークアップ言語（ＸＡＭＬ）である請求項１に記載のシステム。
【請求項４】
前記シェーダーは１つ以上のコアを備えた演算処理装置上で実行するように動作可能である請求項１に記載のシステム。
【請求項５】

前記宣言型のプログラミング・モデルは、前記シェーダーによりレンダリングされている効果がどこで入力を得るかを制御するために潜在的な入力が指定されることを可能にするよう動作可能であり、潜在的な入力は、該効果が適用されているユーザー・インターフェース要素をラスタライズすることによって生じる画素ベースのビットマップを含む請求項１に記載のシステム。

【請求項６】

前記宣言型のプログラミング・モデルは１つ以上の依存特性がシェーダー入力に結合されることを可能にするよう動作可能である請求項１に記載のシステム。

【請求項７】

前記宣言型のプログラミング・モデルは、前記シェーダーを実行することによりブラシとして作用する効果ブラシが定義されることを可能にするよう動作可能である請求項１に記載のシステム。

【請求項８】

前記宣言型のプログラミング・モデルは効果が入力をどのように変えるのかをモデル化するよう動作可能である請求項１に記載のシステム。

【請求項９】

効果が入力をどのように変えるのかについてのモデル化は、特定の画素上で前記シェーダーを実行することにより実行される請求項８に記載のシステム。

【請求項１０】

前記宣言型のプログラミング・モデルは複数パス効果が指定されることを可能にするよう動作可能である請求項１に記載のシステム。

【請求項１１】

シェーダーベースの効果を定義し宣言型のプログラミング・モデルにおいて前記シェーダーベースの効果を使用する方法であって、

ソフトウェア・アプリケーションに対してグラフィック効果をレンダリングするために、宣言的に指定されたシェーダーベースの効果をプログラムによって例示化するステップと、

シェーダーについてグラフィック効果カスタマイゼーションを含む、前記ソフトウェア・アプリケーションのための宣言文にアクセスするステップであって、前記宣言文は宣言型言語で書かれ、前記宣言型言語は拡張マークアップ言語又は拡張アプリケーションマークアップ言語を含む、ステップと、

前記グラフィック効果カスタマイゼーションを備えた前記ソフトウェア・アプリケーションについて前記シェーダーがグラフィック効果をレンダリングすることを可能にするために、シェーダー・プロセッサーへ前記宣言文を送信するステップであって、前記宣言文はブラシが前記シェーダーへの二次的な入力として使用されることを可能にし、潜在的な入力が前記シェーダーへの入力として使用されることを可能にし、潜在的な入力は、効果が適用されているユーザー・インターフェース要素をラスタライズすることによって生じる画素ベースのビットマップを含む、ステップと

を備える方法。

【請求項１２】

前記宣言文が前記シェーダーによって必要とされるときに、前記宣言文は前記シェーダーの動作の全体にわたってアクセスされる請求項１１に記載の方法。

【請求項１３】

前記宣言文は、前記シェーダーが実行される演算処理装置を前記グラフィック効果カスタマイゼーションが利用することを可能にする請求項１１に記載の方法。

【請求項１４】

前記シェーダーは、画素シェーダー、頂点シェーダー及び形状シェーダーからなるグループから選択される種類のシェーダーである請求項１１に記載の方法。

【請求項１５】

複数パス動作中に実行されるべきシェーダーの組を制御しカプセル化する複数パス効果

として宣言型のプログラミング・モデルを利用する方法であって、
拡張マークアップ言語又は拡張アプリケーションマークアップ言語を含む宣言型言語で書かれた宣言文を使用して、複数パス効果クラスから導出される、複数パス効果のためのカスタム方法を呼び出すステップと、
前記カスタム方法が前記複数パス効果の動作を制御することを可能にするために、前記カスタム方法がシェーダー・コンテキスト及び制御情報にアクセスすることを可能にするコンテキストを前記カスタム方法に提供するステップと
を備える方法。
【請求項１６】
前記カスタム方法における複数パス効果中のすべてのパスについて前記グラフィック効果カスタマイゼーションが構成スレッド上で呼び出される請求項１５に記載の方法。
【請求項１７】
前記カスタム方法は適用シェーダー方法を無効にし、前記適用シェーダー方法は、前記カスタム方法における複数パス効果の各パスについてそれぞれのシェーダーをプログラムによって選択するためのロジックを含む請求項１５に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、宣言型プレゼンテーション・フレームワークのためのシェーダーベースの拡張に関する。
【背景技術】
【０００２】
[0001]グラフィック処理装置（ＧＰＵ）はコンピューターのための専用のグラフィック・レンダリング装置である。今日のＧＰＵはコンピューター・グラフィックの操作及び表示において効率的であり、それらの並列構造は、一連の複雑なアルゴリズムのための最も汎用の中央処理装置よりもそれらを有効なものにする。ＣＰＵはまた複数のプロセッサー・コアを有してもよい。ＧＰＵのように、マルチコアＣＰＵは、複数の動作を並行して実行することができる。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
[0002]今日、ソフトウェアを書くソフトウェア開発者は、自身が書くソフトウェア・アプリケーションにおいてこの処理能力を利用したいと考えている。例えば、多くのグラフィック効果又はマルチメディアを利用するデスクトップベース又はウェブベースのアプリケーションなどの、大量の処理リソースを必要とするアプリケーションにおいて、コンピューター中に存在するＧＰＵ又はマルチコア・プロセッサーの能力を利用することは有用である。しかし、多くのプラットフォームはＧＰＵ又はマルチコア・プロセッサーのリソースを利用し得るが、末端の開発者は、通常、今日多くのコンピューターに含まれる余分な処理能力を利用することができるプラットフォーム上で動作するアプリケーションを容易に構築することができない。
【課題を解決するための手段】
【０００４】
[0003]シェーダーを宣言的に制御するための様々な技術及び技法が開示される。宣言型のプログラミング・モデルは、宣言型プレゼンテーション・フレームワークにおけるシェーダーのインスタンス化を制御する宣言文（declarative statements）を使用することを可能にする。宣言的に指定されたシェーダーは、ソフトウェア・アプリケーションのためにグラフィック効果をレンダリングする（ソフトウェア・アプリケーションにグラフィック効果を与える）ためのプレゼンテーション・フレームワークによって後にインスタンス化される（インスタンスを作成される、instantiated）。１つの実施例において、宣言型のプログラミング・モデルは、単一パス動作（single pass operation）中に実行される

べきシェーダーを制御しカプセル化する単一パスシェーダーとして使用することができる。別の実施例において、複数パス動作（multiple pass operation）中に実行されるべき１組のシェーダーを制御しカプセル化する複数パス効果として宣言型のプログラミング・モデルを利用するための方法が記載される。カスタム方法（カスタム・メソッド、custom method）が複数パス効果について呼び出される。カスタム方法は複数パス効果クラスから導かれたものである。カスタム方法は、カスタム方法が複数パス効果の動作を制御することを可能にするためのシェーダー・コンテキスト及び制御情報にカスタム方法がアクセスすることを可能にする、コンテキストを備えている。

【０００５】

[0004]この概要は詳細な説明においてさらに以下に述べる概念の選択を単純化された形式で紹介するために提供された。この概要は、特許請求される主題の主な特徴又は本質的な特徴を識別するようには意図されず、特許請求される主題の範囲を決定する際に助けとして使用されるようにも意図されない。

【図面の簡単な説明】

【０００６】

【図１】[0005]画素シェーダーの動作を制御するためのシステムの線図である。

【図２】[0006]宣言型プレゼンテーション・フレームワークにおけるシェーダーのインスタンス化及び動作を制御する宣言文が使用されることを可能にする、１つの実施例についての宣言型のプログラミング・モデルの線図である。

【図３】[0007]宣言型のプログラミング・モデルを使用してシェーダーベースの効果を定義することに関与する段階を示す１つの実施例についての処理フロー図である。

【図４】[0008]シェーダーへの入力として使用されているブラシ（brush）を示す１つの実施例についての線図である。

【図５】[0009]シェーダーへの二次的な入力及びシェーダーへの潜在的な（implicit）入力としてブラシの使用を示す１つの実施例についてのある例示的なソース・コードである。

【図６】[0010]シェーダー入力に対する特性の結合（binding）を示す１つの実施例についての線図である。

【図７】[0011]シェーダー入力に対する特性の結合を示す１つの実施例についての例示的なソース・コードである。

【図８】[0012]効果ブラシ（effect brush）を定義する際のシェーダーの使用を示す１つの実施例についての線図である。

【図９】[0013]効果ブラシを定義する際のシェーダーの使用を示す１つの実施例についての例示的なソース・コードである。

【図１０】[0014]シェーダーによる一般的な変換の使用による及び／又は画素自体の上でシェーダーを実行することによるヒット・テスト（hit testing）を示す１つの実施例についての線図である。

【図１１】[0015]シェーダーによる一般的な変換の使用によるヒット・テストを示す１つの実施例についての例示的なソース・コードである。

【図１２】[0016]複数パス動作中に実行されるべき１組のシェーダーを制御しカプセル化する複数パス効果を示す１つの実施例についての処理フロー図である。

【図１３】[0017]ユーザー・インターフェース・スレッドから構成スレッドへの制御の流れを示す１つの実施例についての線図である。

【図１４】[0018]１つの実施例のコンピューター・システムの線図である。

【発明を実施するための形態】

【０００７】

[0019]本明細書において技術及び技法は宣言型のプログラミング・モデルを備えたシェーダーを使用するための技術及び技法として一般的なコンテキストにおいて述べられてもよいが、技術及び技法はまた、これらに加えて他の目的に役立つ。１つの実施例において、本明細書に記載される１つ以上の技術は、ＭＩＣＲＯＳＯＦＴ（登録商標）．ＮＥＴフ

レームワークなどのソフトウェア開発プラットフォーム内の機能、又は開発者がソフトウェア・アプリケーションを開発し及び／又はカスタマイズすることを可能にするためのプラットフォームを提供する他の種類のプログラム又はサービスからの機能として実施することができる。

【０００８】

[0020]、宣言型プレゼンテーション・フレームワーク上で実行される宣言型のプログラミング・モデル１０８を使用してシェーダー１０６の動作を制御するためのシステム１００の線図が図１に示される。本明細書において使用される「宣言文」なる語及び「宣言的に」なる語は、関連する宣言型のフレームワークの基本的な実施に対してその実行を委ねる、テキストベースのマークアップ又は命令コード（imperative code）を含むことを意図する。宣言型言語の例は、少数の限定的でない例を挙げると、ＸＡＭＬ及びＸＭＬを含む。本明細書において使用される「宣言型のプログラミング・モデル」なる語は、関連する宣言型のフレームワークによって処理されるべき効果をインスタンス化し使用するために、開発者、別のユーザー又はプログラムが当該プログラミング・モデルによって宣言文を設定する、プログラミング・モデルを含むことを意図する。本明細書において使用される「宣言型プレゼンテーション・フレームワーク」なる語は、宣言型のプログラミング・モデルを使用して宣言文によって開発者、別のユーザー又はプログラムによって提供される、より抽象的な記述とのインタラクションを表示し可能にするシステムを含むことを意図する。宣言型プレゼンテーション・フレームワークの制限的でない１つの例は、ＭＩＣＲＯＳＯＦＴ Ｗｉｎｄｏｗｓ（登録商標） Ｐｒｅｓｅｎｔａｔｉｏｎ Ｆｏｕｎｄａｔｉｏｎ（すなわち、ＷＰＦ）であり、これは、（バージョン３．０で始まる）ＭＩＣＲＯＳＯＦＴ．ＮＥＴフレームワークのグラフィカルなサブシステム機能である。ＷＰＦは、アプリケーションを構築するための一貫したプログラミング・モデルを提供し、ユーザー・インターフェース・ロジックとビジネスロジックとの間の明確な分離を提供する。宣言型プレゼンテーション・フレームワークの制限的でない別の例はＭＩＣＲＯＳＯＦＴ Ｓｉｌｖｅｒｌｉｇｈｔである。

【０００９】

[0021]本明細書において使用される「シェーダー（shader）」なる語は、主としてグラフィカルなレンダリング効果を行うためにグラフィックのリソースによって使用される１組のコンピューター命令を含むことを意図する。システム１００に利用することができる、画素シェーダー、頂点（vertex）シェーダー及び／又は形状（配置、幾何学、geometry）シェーダーなどの異なる種類のシェーダーが存在する。頂点シェーダーは一連の頂点に影響を及ぼし、したがって、位置、色及びテクスチャ座標のような頂点特性を変更することができる。頂点シェーダーによって計算された頂点は、通常、形状シェーダーに渡される。形状シェーダーは、メッシュから頂点を追加及び削除することができる。形状シェーダーは、手続きに従って形状（配置）を生成するため又は既存のメッシュに容積測定の（volumetric）詳細を加えるために使用することができ、これは、通常は、演算処理装置上で処理するにはコストがかかりすぎる。画素シェーダーはフラグメントシェーダーとしてより一般に知られており、頂点シェーダー及び形状シェーダーによって生成された多角形がラスターライズされる場合、個々の画素の明度（color value）を計算する。画素シェーダーは、通常、シーン照明（scene lighting）、並びにバンプ・マッピング及び色調整などの関連する効果のために使用される。

【００１０】

[0022]１つ以上の演算処理装置１０４上でシェーダー１０６が実行される。演算処理装置１０４の少数の制限的でない例はグラフィック処理装置（ＧＰＵ）又はマルチコア・プロセッサーを含み得る。シェーダー１０６が１つ以上の演算処理装置１０４上で実行される場合、「シェーダー・プロセッサー」なる語は演算処理装置１０４上でのシェーダーの実行を指すのに使用される。以前に述べたように、シェーダー１０６は所与のソフトウェア・アプリケーションに対して効果を与える（レンダリングする）責務を有する。宣言型のプログラミング・モデル１０８は、開発者又は他のユーザーがユーザー宣言型の（user

-declared）シェーダー・カスタマイゼーション１１０を提供することを可能にする。図２－１３のフロー及びコードの例においてここでさらに詳細に述べられるように、シェーダー・カスタマイゼーション１１０は、ユーザーがシェーダー１０６のインスタンス化及び／又は動作を制御することを可能にする。

【００１１】

[0023]図２は、宣言型プレゼンテーション・フレームワークにおけるシェーダーのインスタンス化及び動作を制御する宣言文が使用されることを可能にする、１つの実施例についての宣言型のプログラミング・モデル１５０の線図である。宣言型のプログラミング・モデル１５０は、シェーダーの制御のために宣言的に指定することができる様々な機能をサポートする。１つの実施例において、宣言型のプログラミング・モデル１５０は、単一パス動作中に実行されるべきシェーダーを制御しカプセル化する単一パスシェーダーとして使用することができる。別の実施例において、宣言型のプログラミング・モデル１５０は、複数パス動作中に実行されるべき１組のシェーダーを制御しカプセル化する複数パス効果として使用することができる。複数パス効果は、本明細書における複数パス効果１６４についての議論及び図１２の議論においてより詳細に述べられる。

【００１２】

[0024]宣言型のプログラミング・モデル１５０を使用して、ブラシは、シェーダーへの入力１５２として指定することができる。本明細書において使用される「ブラシ」なる語は、形状を満たすために適用されると、どの画素位置にどの色を配置するかを決定することができるオブジェクトを表すことを意図する。より抽象的には、ブラシは２Ｄ空間から色へのマッピング機能を表す。例示的なブラシは、単純なラスター画像、線形且つ放射状の勾配（グラディエント）、ライブビデオのソース、他のベクトル・グラフィック・レンダリングからのソースなどを含む。ブラシはその上画素シェーダーに対する入力としての役割を果たし、シェーダーによって処理されるべき「サンプラー」としてシェーダー自体に対して提示することができる。シェーダーへの入力としてブラシを利用することについてのさらなる詳細は図４－５についての議論において提供される。

【００１３】

[0025]宣言型のプログラミング・モデル１５０は、潜在的な入力がシェーダー・プロセッサーに提供されることを可能にして、グラフィック効果がその一次入力などの入力１５４をどこで得るかを制御する。本明細書において使用される「潜在的な入力（implicit input）」なる語は、「効果／シェーダーが適用されているユーザー・インターフェース要素をラスタライズすることに起因する、画素ベースのビットマップ」を含むよう意図する。したがって、例えば、効果がボタンに適用される場合、「潜在的な入力」はそのボタンを表すビットマップである。潜在的な入力の一例は、図５においてより詳細に述べられる。

【００１４】

[0026]シェーダー入力１５６に対する特性結合（property binding）はまた、宣言型のプログラミング・モデル１５０によりサポートされる。本明細書において使用される「特性結合」なる語は、データ結合を行うかそうでなければ柔軟な制御を行うときにアニメーション実行するために参照することができるオブジェクト指向のクラス定義の宣言型の特性を含むことを意図する。。いくつかの実施例において、そのような特性は依存特性（Dependency Properties）として知られている。本明細書において使用される「シェーダー入力」なる語は、レジスターインデックスを介してシェーダーにとって利用可能になる「シェーダー定数」及び「シェーダーサンプラー」の両方を含むことを意図する。本明細書において使用される「シェーダー定数」なる語は、シェーダープログラムの実行の全体にわたって一定のままである、シェーダープログラムに提供される値を含むことを意図する。シェーダー定数についての少数の制限的でない例は、浮動小数点値、色、点、ベクトル、行列などを含む。本明細書において使用される「シェーダーサンプラー」なる語は、特定の座標点においてシェーダープログラム内でサンプリングすることができるシェーダープログラムに提供される多次元アレイを含むことを意図する。少数の制限的でない例は２

Ｄビットマップ及び３Ｄテクスチャー・ボリュームを含んでもよい。言いかえれば、宣言型のプログラミング・モデル１５０は、依存特性がシェーダー入力と宣言的に接続されることを可能にする。シェーダー入力１５６に対する特性結合は、図６及び７においてより詳細に述べられる。１つの実施例において、シェーダー入力に対する特性結合をサポートすることによって、それらの依存特性は、シェーダーの実施についてさらに知る必要のあるシェーダーを（特性宣言を含むより高いレベルのオブジェクトを介して）使用している開発者なしにシェーダーに渡すことができる。

【００１５】

[0027]宣言型のプログラミング・モデル１５０はまた、効果ブラシがシェーダーの実行からのブラシとして作用するように定義される（１５８）ことを可能にする。本明細書において使用される「効果ブラシ」なる語は、シェーダー又は一連のシェーダーの実行による２Ｄ空間から色へのその機能を定義する（上に定義されるような）あらゆる「ブラシ」を含むことを意図する。効果ブラシは、一般に、他のブラシと同様の方法で使用することができる。１組の明示的な入力を仮定すると、効果ブラシは、ブラシが適用されている目標を満たすために色を生成する。この機能を利用して、システムはシェーダーを実行するために予め何もレンダリングする必要はない。言いかえれば、効果ブラシ機能は、シェーダーがユーザー・インターフェース要素のブラシとして実行することを可能にする。効果ブラシは図８及び９においてより詳細に述べられる。

【００１６】

[0028]宣言型のプログラミング・モデル１５０によってサポートされる別の機能は、効果が入力をどのように変えるかをモデル化するための、一般的な変換１６０を使用するヒット・テスト及び座標空間変換である。本明細書において使用される「ヒット・テスト」なる語は、シェーダーが適用された後に座標空間における特定の点の位置を求める要求の処理を含むことを意図する。本明細書において使用される「座標空間変換」なる語は、シェーダーベースの効果を経る必要性及びシェーダーが潜在的に画素を変換するのと同じ方法で座標を変換するコードを有することを含む、変換階層に沿った任意のマッピングを含むことを意図する。インタラクトされるときにマウス位置又は入力ポインター位置を効果が適用される変換の逆を介して変換する必要があるように、ある効果はそのコンテンツをあちこち移動させることができる。一般的な変換１６０機能を使用するヒット・テストは、ある効果が入力をモデル化されるようにシフトする方法を可能にする。例えば、効果がボタンを「回転させる（swirl）」場合、入力ポインターは、それが回転されたボタンの表現の上にあり、且つ、ボタンがもともと存在していたがもはやそこに存在していない空間の上にある場合でないときに、ボタンの「上に」あるものとして登録するべきである。ヒット・テストは図１０及び１１においてより詳細に述べられる。

【００１７】

[0029]宣言型のプログラミング・モデル１５０はまた、画素自体１６２の上でシェーダーを実行することによりヒット・テストをサポートする。この機能により、シェーダーは、実際の画素座標を得るために特定の画素上で実行することができる。これは、上述のようなヒット・テストに必要な座標空間マッピングを行うための１つの可能な実施例を表す。これは図１０においてより詳細に述べられる。

【００１８】

[0030]複数パス効果１６４はまた宣言型のプログラミング・モデル１５０によりサポートされる。複数パス効果により、複数のレベルの効果があり得、一度に１つのシェーダー上で起こる１組の動作として具体化される。開発者は、宣言型のプログラミング・モデル１５０を使用して、シェーダーの複数のパスを制御し拡張することができる。複数パス効果１６４は、図１２においてより詳細に述べられる。

【００１９】

[0031]ここで図３－１３に移ると、シェーダー動作を制御するためのシステムの１つ以上の実施例を実施するための段階がより詳細に記述される。いくつかの実施例において、図３－１３のプロセスは、（図１４の）計算装置５００の動作論理において少なくとも部

分的に実施される。

【００２０】

[0032]図３は、宣言型のプログラミング・モデルを使用してシェーダーベースの効果を定義することに関与する段階を示す１つの実施例についての処理フロー図２００である。宣言的に参照されたシェーダーベースの効果は、ソフトウェア・アプリケーションにグラフィック効果を与える（レンダリングする）ための宣言型プレゼンテーション・フレームワークによってプログラムで例示化される（instantiated）（段階２０２）。シェーダーの少数の制限的でない例は、画素シェーダー、頂点シェーダー及び／又は形状シェーダーを含んでもよい。宣言文はソフトウェア・アプリケーションについてアクセスされる（段階２０４）。言いかえれば、シェーダーのためのグラフィック効果カスタマイゼーションを含む宣言文は、宣言文を含む１つ以上のファイルから取り出される。１つの実施例において、それらがシェーダーによって必要とされるとき、宣言文はシェーダーの動作の全体にわたってアクセスされる（段階２０４）。グラフィック効果カスタマイゼーションによりソフトウェア・アプリケーションに対してシェーダーがグラフィック効果を与えることを可能にするために、宣言文はシェーダー・プロセッサーへ送られる（段階２０６）。言いかえれば、宣言文はシェーダー・プロセッサーへ送られ、宣言型の方法で開発者、他のもの又はプログラムのユーザーによって指定された所望のグラフィック効果カスタマイゼーションを実行するためにシェーダーによって使用される。１つの実施例において、宣言文は、シェーダーが実行されるＧＰＵやマルチコア・プロセッサーなどの演算処理装置（の能力）をグラフィック効果カスタマイゼーションが利用することを可能にする。

【００２１】

[0033]図４は、シェーダー２２２への入力として使用されている１つ以上の宣言文２２４を使用して定義されたブラシ２２６を示す１つの実施例についての線図２２０である。ブラシは、放射状の勾配、ビットマップ画像、動画、ベクトル・グラフィックなどの特徴が、（いくつかの既存のシェーダーがサポートする正当なビットマップとは対照的な）シェーダーへの入力として役立つことを可能にする。ブラシの概念をより詳細に示すためのコードの例は図５に示される。

【００２２】

[0034]図５は、シェーダーへの二次的な入力としてブラシの使用を示し、シェーダーへの潜在的な入力の使用を示す、１つの実施例についてのある例示的なソース・コード２５０である。示された例において、ｍｙＢｕｔｔｏｎと呼ばれるボタンが宣言的に指定されている。当該ボタンは、テクスチャー減算効果（texture subtraction effect）の最下段特性（底特性）についての入力としてブラシ２５４を含むボタン効果により宣言されたものである。テクスチャー減算効果の頂点特性は、効果が適用されている要素のビットマップ・レンダリングーこの場合、ｍｙＢｕｔｔｏｎのレンダリングーを表す潜在的な入力特性２５２を含む。最終的な結果は、当該ボタンが、今は、それから減算されたブラシ２５４からの画像を備えた元のボタンのレンダリングのように見えるということである。

【００２３】

[0035]図６は、シェーダー定数及び／又はサンプラーであり得る、１つ以上のシェーダー入力２８４への１つ以上の依存特性２８８の結合を示す１つの実施例の線図２８０である。以前に述べたように、シェーダー入力はレジスターインデックスによってシェーダーにとって利用可能になる。依存特性は、所与のシェーダー２８２のシェーダー入力２８４に結合するために宣言文２８６を使用して指定することができる。１つの実施例において、依存特性をシェーダー入力と接続することによって、依存特性は、定義することができ、シェーダーの詳細についてユーザーが心配する必要なくシェーダーに渡すことができる。宣言型の方法によるシェーダー入力への（より具体的には、シェーダー定数への）依存特性の結合の例は図７に示される。図７に示される例示的なソース・コードにおいて、シェーダー定数に対する２つの依存性（３０４及び３０６）は宣言的に指定される。それらの依存性（３０４及び３０６）は、シェーダー定数３０８及び３１０などのシェーダー３０２の実際の登録に結合される。１つの実施例において、当該結合はコールバックの使用

を通じて遂行され、これは、依存特性がいつ変更されようとも、効果、及びしたがってシェーダーが実行される次回のための新たな値でシェーダー入力を更新するよう宣言型のフレームワークに命じるコールバック関数が呼び出されることを意味する。

【００２４】

[0036]図８は、効果ブラシを定義する際のシェーダーの使用を示す１つの実施例についての線図３２０である。図２に述べられるように、効果ブラシ３２６は、シェーダー３２２の実行からのブラシとして作用するために宣言文３２４を使用して定義することができる。１組の明示的な入力を考慮すると、効果ブラシ３２６は、ブラシが適用されているその目標を満たすために色を生成する。効果ブラシ機能は、シェーダー３２２がユーザー・インターフェース要素のブラシとして実行されることを可能にする。効果ブラシ３２６を宣言的に指定するための例示的なソース・コード３３０は図９に示される。図９に示される例において、効果ブラシはカスタム効果３３４により宣言的に指定される（３３２）。示された例におけるカスタム結果３３４は、矩形に対してフラクタル・フィル（fractal fill）を生成する。この例において、効果ブラシは、効果（Effect）種類である単一のパラメーター、Ｅｆｆｅｃｔとともに使用される。更に、ＭａｎｄｅｌｂｒｏｔＥｆｆｅｃｔは、シェーダー入力として基本的なシェーダーに渡される３つの依存特性を備えたカスタムＳｈａｄｅｒＥｆｆｅｃｔである。その後、効果の出力は矩形を満たすために使用される。

【００２５】

[0037]ここで図１０に移ると、シェーダー３５２による一般的な変換３５６を使用すること及び／又は特定の画素３５８においてシェーダー３５２を実行することにより、ヒット・テスト及び座標空間変換の性能を示す、１つの実施例についての線図が示される。以前に述べたように、ヒット・テストは、シェーダーが適用された後に座標空間において指定された点の位置を求める要求の処理を意味する。一般的な変換３５６は宣言文３５４を使用して指定することができ、ヒット・テスト及び座標空間変換がモデル化されることを可能にする。一般的な変換の例は図１１の例示的なソース・コード３８０に示される。代替的に又はさらに、ヒット・テストは特定の画素３５８上でシェーダー３５２を実行することにより行うことができる。

【００２６】

[0038]図１２は、複数パス動作中に実行されるべき１組のシェーダーを制御しカプセル化する複数パス効果を示す１つの実施例についての処理フロー図４４０である。本明細書において使用される「複数パス効果」なる語は、所望のレンダリングを達成するために１つより多くのシェーダーを複数回潜在的に呼び出す効果を含むことを意図する。本明細書において使用される「複数パス動作」なる語は、複数のパス及び複数のシェーダーの実行を制御するコードを含むことを意図する。カスタム方法は複数パス効果に対して呼び出される（段階４４２）。１つの実施例において、カスタム方法は複数パス効果クラスから導かれる。１つの実施例において、カスタム方法は、それが適用シェーダー（例えば、ＡｐｐｌｙＳｈａｄｅｒｓ）の方法を無効にするように定義される。適用シェーダーの方法は、カスタム方法における複数パス効果の各々のパスについてそれぞれのシェーダーをプログラムによって選択するためのロジックを含む。

【００２７】

[0039]カスタム方法は、当該カスタム方法が複数パス効果の動作を制御することを可能にする（段階４４６）ために、当該カスタム方法がシェーダー・コンテキスト及び制御情報にアクセスすることを可能にする（段階４４４）コンテキストを備えている。言いかえれば、カスタム方法はシステムによって呼び出され、シェーダーインデックス、現在の地球規模（グローバル・スケール）、シェーダー定数、シェーダーサンプラー及び／又はシェーダーを選択し実行するための能力のような関連情報のアクセス及び制御を可能にする「コンテキスト」を備えている。この機構によって、カスタム方法における複数パス効果の開発者は、複数パス効果レンダリングを制御することができる。

【００２８】

[0040]１つの実施例において、複数パス効果機能の使用によって、複数シェーダー効果ベースクラス（base class）の特定のサブクラスのユーザーは、宣言型のグラフィック又はユーザー・インターフェース表現にその効果を含めるために、当該効果を宣言的に指定することのみを必要とする。

【００２９】

[0041]１つの実施例において、グラフィック効果カスタマイゼーションは、所与の複数パス効果におけるすべてのパスについて構成スレッド上で呼び出される。構成スレッドは図１３においてより詳細に述べられ、それはここで次に議論される。

【００３０】

[0042]図１３は、ユーザー・インターフェース・スレッドから構成スレッドまでの制御の流れを示す１つの実施例についての線図４８０である。本明細書において使用される「ユーザー・インターフェース・スレッド」なる語は、アプリケーションのユーザー・インターフェース要素（ボタン、テキストボックス、リスト）が応答する実行の主要なスレッドを含むことを意図する。本明細書において使用される「構成スレッド」なる語は、ディスプレイのレンダリングを制御するユーザー・インターフェース・スレッドとは異なるスレッドを含むことを意図する。構成スレッドに対する動作はＵＩスレッドに対する動作によってブロックされない。

【００３１】

[0043]図１３に戻ると、フローの制御はユーザー・インターフェース・スレッドと構成スレッドとの間で示される。ユーザー・インターフェース・スレッドにおいては、宣言文が解析され、効果が構築され（段階４８２）、次に、効果を含む視覚的なツリーが構築される（段階４８４）。構成スレッドにおいては、構成側の視覚的なツリーに相当するものが構築される（段階４８６）。ユーザー・インターフェース・スレッドに戻ると、効果依存特性は結合され評価される（段階４８８）。ユーザー・インターフェース・スレッドに存在する間、ユーザー・インターフェース・スレッド・レンダリング・ループは、アニメーションを評価し、構成レンダリングを呼び出す（段階４９０）。その後、構成スレッドはフレームをレンダリングし（段階４９２）、効果により視覚的なツリーノード上でヒットをレンダリングする（段階４９４）。１つの実施例において、次に起こることは、レンダリングされる効果の種類に基づいて異なる（段階４９６）。マルチパス効果の場合には、効果の実行を制御するために、ＡｐｐｌｙＳｈａｄｅｒｓ方法が構成スレッドから呼び出される。単一パス効果の場合には、構成スレッドが如何なるユーザー・コードも呼び出すことなく効果を単に実行するのに十分な情報が、構成スレッド上に既に存在する。

【００３２】

[0044]図１４に示されるように、システムの１つ以上の部分を実施するのに使用するための例示的なコンピューター・システムは、計算装置５００などの計算装置を含む。そのほとんどの基本構成において、計算装置５００は、通常、少なくとも１つの演算処理装置５０２及びメモリー５０４を含む。計算装置の正確な構成及び種類によって、メモリー５０４は、（ＲＡＭなどの）揮発性のものであってもよいし、（ＲＯＭ、フラッシュ・メモリーなどのように）不揮発性のものであってもよいし、又はこれら２つのある組合せであってもよい。この大部分の基本構成は破線５０６によって図１４に示される。

【００３３】

[0045]さらに、装置５００はまた、追加の特徴／機能を有してもよい。例えば、装置５００はまた、磁気もしくは光のディスク又はテープを含むがこれらに限定されない、（取り外し可能及び／又は取り外し不能な）追加の記憶装置を含んでもよい。そのような追加の記憶装置は、取り外し可能な記憶装置５０８及び取り外し不能な記憶装置５１０によって図１４に示される。コンピューター記憶媒体は、コンピューター読み取り可能な命令、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の記憶のための任意の方法又は技術において実施される、揮発性及び不揮発性、取り外し可能及び取り外し不能な媒体を含む。メモリー５０４、取り外し可能な記憶装置５０８及び取り外し不能な記憶装置５１０は、すべて、コンピューター記憶媒体の例である。コンピューター記憶媒体は、Ｒ

ＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリーもしくは他のメモリー技術、ＣＤ－ＲＯＭ、ディジタル・バーサタイル・ディスク（ＤＶＤ）もしくは他の光学記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、又は所望の情報を格納するために使用することができ且つ装置５００によってアクセスできる任意の他の媒体を含むがこれらに限定されない。如何なるそのようなコンピューター記憶媒体も装置５００の一部であり得る。

【００３４】

[0046]計算装置５００は、計算装置５００が他のコンピューター／アプリケーション５１５と通信することを可能にする１つ以上の通信接続５１４を含む。装置５００はまた、キーボード、マウス、ペン、音声入力装置、タッチ入力装置などの入力装置５１２を有してもよい。ディスプレイ、スピーカー、プリンターなどの出力装置５１１も含まれてもよい。これらの装置は当技術分野において周知であり、ここで詳細に議論する必要はない。

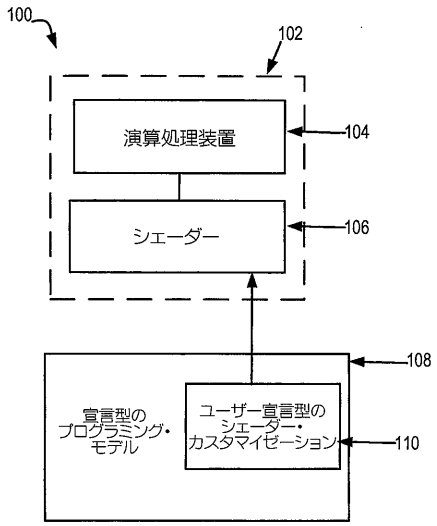
【００３５】

[0047]主題は構造的特徴及び／又は方法論の作用に特有の言語で説明されたが、添付の特許請求の範囲において規定される主題が上述の特定の特徴又は作用に必ずしも限定されないことが理解されるべきである。そのようなものではなく、上述の特定の特徴及び作用は、請求項を実行する例示的な形式として開示されたものである。本明細書に記載されるような実施例及び／又は以下の特許請求の範囲によって記載されるような実施例の趣旨にあるすべての均等物、変更及び修正は保護されることが望まれる。

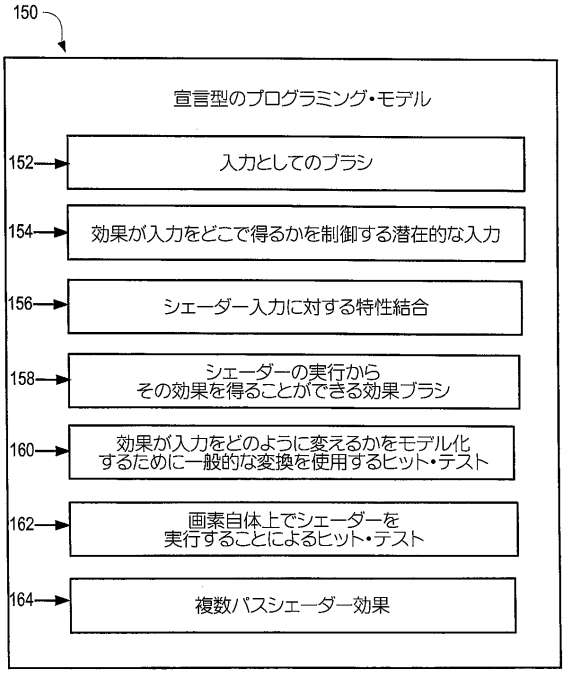
【００３６】

[0048]例えば、コンピューター・ソフトウェア技術における当業者であれば、本明細書において記載された例が、実施例において表現されたようなものより少ない又は追加のオプション又は機能を含むように、１つ以上のコンピューター上で異なったように組織化することができることを認識するであろう。

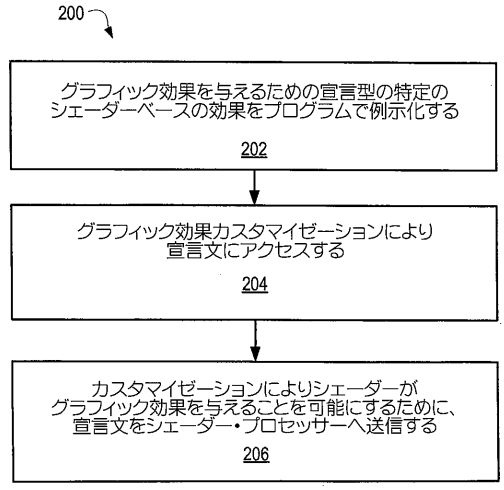
【図１】



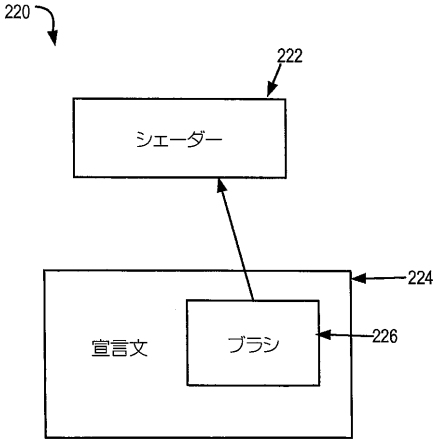
【図２】



【図３】



【図４】



【図５】

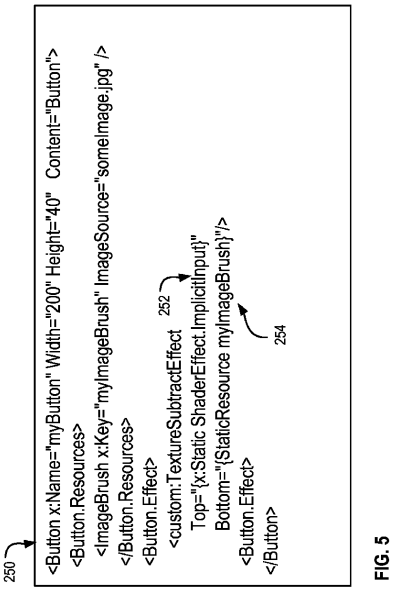
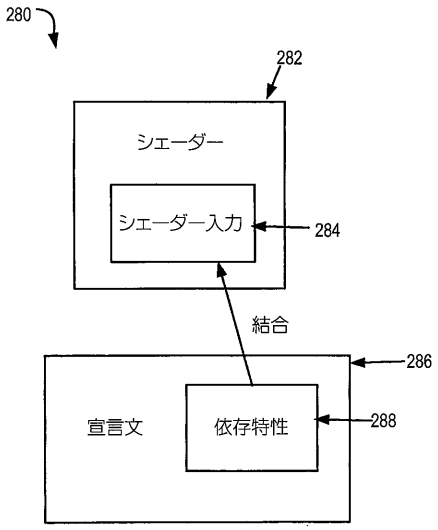
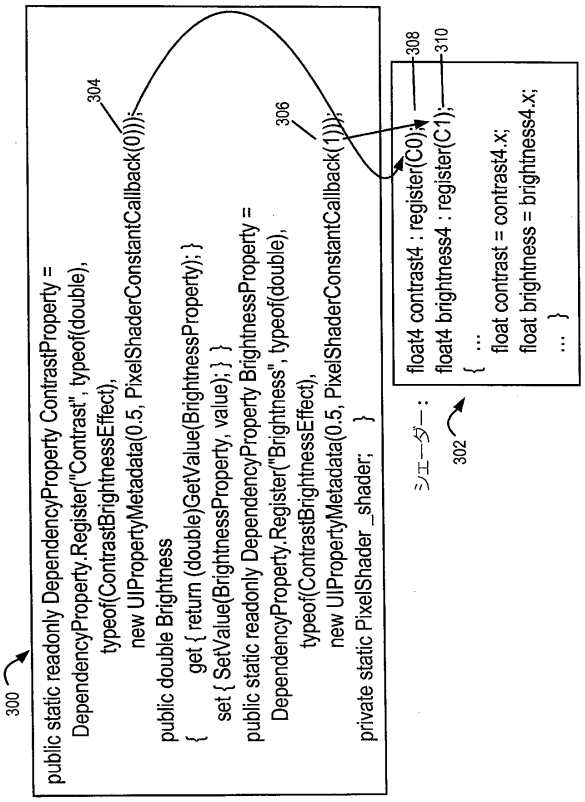


FIG. 5

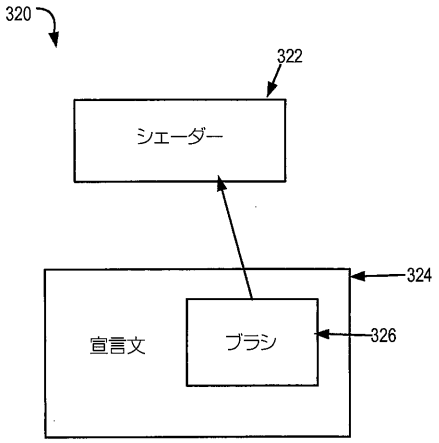
【図６】



【図７】



【図８】



【図９】

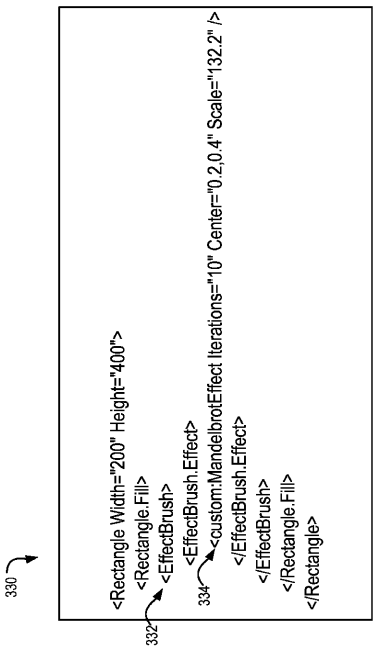
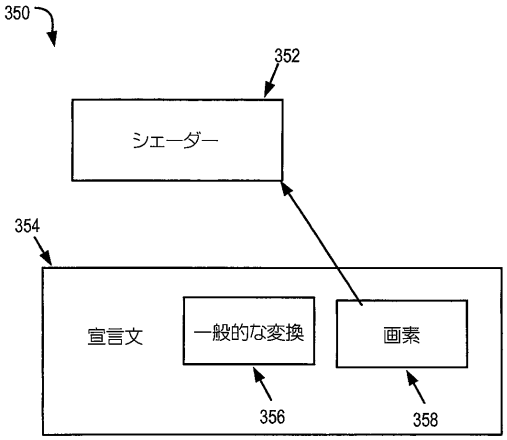


FIG. 9

【図１０】



【図１１】

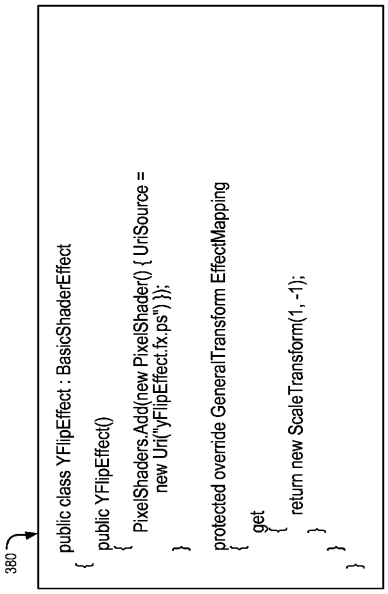
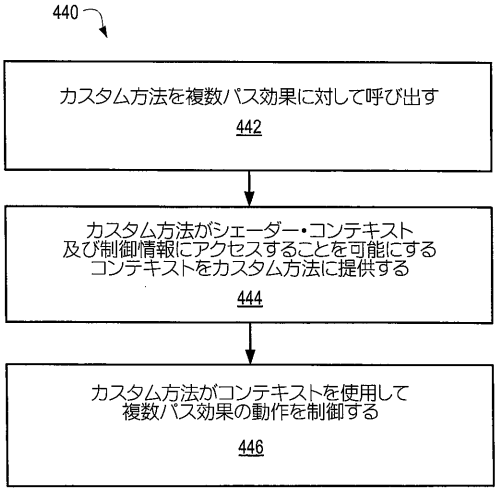
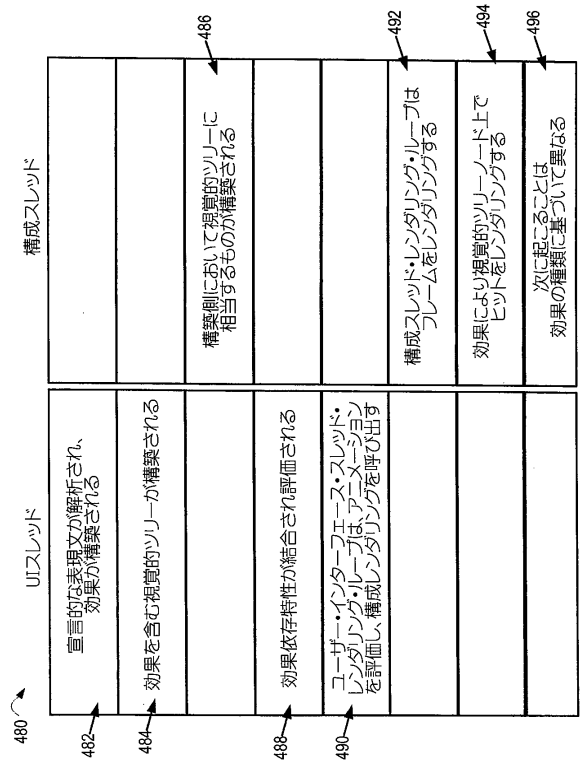


FIG. 11

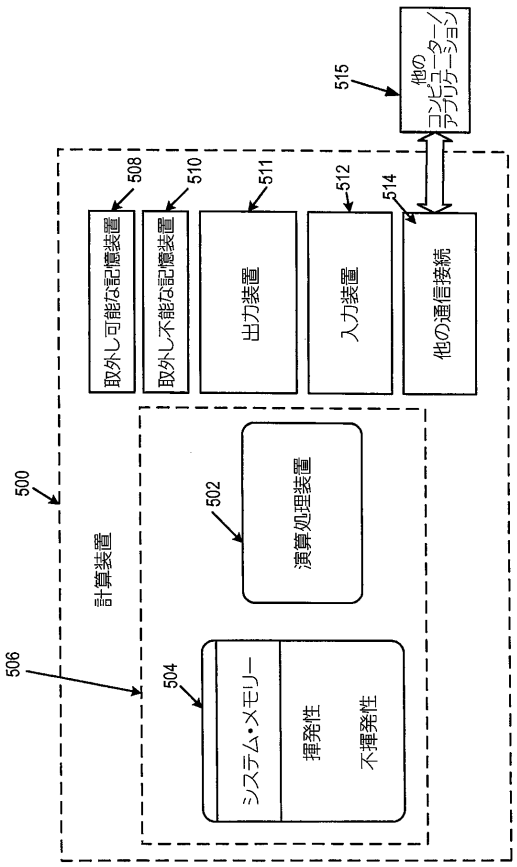
【図１２】



【図１３】



【図１４】



フロントページの続き

(51)Int.Cl. FI
G09G 5/36 530X
G09G 5/00 550H

(74)代理人 100147991
弁理士 鳥居 健一
(72)発明者 シェヒター, グレッグ・ディー
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 コーポレーション, エルシーエイ－インターナショナル・パテンツ
(72)発明者 シュナイダー, ゲルハルト
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 コーポレーション, エルシーエイ－インターナショナル・パテンツ
(72)発明者 ミハイル, アシュラフ・エイ
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 コーポレーション, エルシーエイ－インターナショナル・パテンツ
(72)発明者 クラーク, ブレンダン
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 コーポレーション, エルシーエイ－インターナショナル・パテンツ

審査官 千葉 久博

(56)参考文献 特開２００７－１７２４５４（ＪＰ, Ａ）
特開２００５－３２２２２４（ＪＰ, Ａ）
特開２００５－０７１３６８（ＪＰ, Ａ）
特表２００７－５３６６２２（ＪＰ, Ａ）
特表２００５－５２１１７９（ＪＰ, Ａ）
特表平０９－５０１７８６（ＪＰ, Ａ）
国際公開第２００７／００５７３９（WO, Ａ１）

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｔ １５／００－１５／８７
Ｇ０６Ｔ １／００－１／２０
Ｇ０９Ｇ ５／００－５／４２
Ｇ０６Ｆ ３／０１, ３／０４８
Ｇ０６Ｔ １３／００－１３／８０, １９／００, １９／２０